

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Eva Ondrášková

OracleVisualTools

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. David Hoksza

Studijní program: Informatika, Obecná informatika

2009

Ráda bych na tomto místě poděkovala vedoucímu mé bakalářské práce RNDr.Davidu Hokszoovi za vedení mé práce. Dále děkuji Ing. Davidu Krchovi, Technology Sales Consultant Oracle Czech, za konzultace a vstřícný přístup. A také závěrem patří velké poděkování mým rodičům za trpělivost a morální podporu.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce

V Praze dne 11.12.2009

Eva Ondrášková

Název práce: Oracle Visual Tools
Autor: Eva Ondrášková
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. David Hoksza
e-mail vedoucího: hoksza@ksi.mff.cuni.cz

*Abstrakt: Cílem bakalářské práce bylo vytvořit pro produkt Oracle Database sadu uživatelsky orientovaných grafických nástrojů pro Export, Import a SQL*Loader a umožnit tak jednoduchý přístup k funkcionalitě těchto nástrojů, která je dostupná pouze na příkazové řádce. Dále pak umožnit ladění SQL příkazů vytvořením monitorovacího nástroje Profiler pro online monitorování SQL příkazů spouštěných v databázi. Program byl vypracován v jazyku C# pomocí nástroje Microsoft Visual Studio 2008 a odladěn na verzi OracleDatabase 10g Express Edition.*

Klíčová slova: (Oracle,databáze, Export,Import,SQL*Loader,Profiler)

Title: Oracle Visual Tools
Author: Eva Ondrášková
Department: Department of Software Engineering
Supervisor: RNDr. David Hoksza
Supervisor's e-mail address: hoksza@ksi.mff.cuni.cz

*Abstract: The main goal of the bachelor thesis was to create for the product of the Oracle Database a set of user-oriented graphical tools for Export, Import and SQL*Loader offer an easy acces to functionality of these tools that is available only on the command line and then to enable an easier tuning of SQL statements creating monitoring tool Profiler that facilitates the online monitoring of the statements lanuched in the database. The program was created in C# language with Microsoft # Visual Studio 2008 tool and debugged in OracleDatabase 10g Express Edition.*

Keywords: (Oracle, database, Export, Import, SQL*Loader, Profiler)

OBSAH

1.	ÚVOD	5
2.	ANALÝZA	7
2.1.	PŘEHLED APLIKACÍ ZABÝVAJÍCÍCH SE TEMATIKOU MANIPULACE S DATY A MONITOROVÁNÍM SQL PŘÍKAZŮ	8
2.2.	EXPORT, IMPORT, SQL*LOADER	9
2.3.	PROFILER	11
2.4.	DISKUZE K ŘEŠENÍ PROFILERU	14
3.	DESIGN ŘEŠENÍ	17
3.1.	PŘEHLED ARCHITEKTURY KOMUNIKACE ORACLEVISUALTOOLS	17
3.2.	EXPORT, IMPORT, SQL*LOADER	18
3.3.	PROFILER	20
4.	PROGRAMÁTORSKÁ DOKUMENTACE	25
4.1.	WINDOWS FORMS	25
4.2.	HLAVNÍ TŘÍDA	25
4.3.	PŘIPOJENÍ K DATABÁZI	25
4.4.	ČTENÍ HODNOT Z REGISTRU	26
4.5.	SOUBORY A ADRESÁŘE	26
4.6.	EXPORT, IMPORT, SQL*LOADER	27
4.7.	PROFILER	29
5.	UŽIVATELSKÁ DOKUMENTACE	35
5.1.	ZÁKLADNÍ FUNKCE PROGRAMU	35
5.2.	MINIMÁLNÍ KONFIGURACE	35
5.3.	NUTNÁ OPRÁVNĚNÍ	35
5.4.	INSTALACE	35
5.5.	SPUŠTĚNÍ APLIKACE	36
5.6.	PŘIPOJENÍ K DATABÁZI	37
5.7.	ODPOJENÍ OD DATABÁZE	40
5.8.	ZMĚNA PRACOVNÍHO ADRESÁŘE	40
5.9.	OVLÁDÁNÍ NÁSTROJŮ	40
6.	TESTOVÁNÍ APLIKACE	50
6.1.	STRUČNÝ POPIS SÍŤOVÉHO PROSTŘEDÍ	50
6.2.	SCHÉMA TESTOVACÍHO PROSTŘEDÍ	51
6.3.	POPIS INSTALACE NA HOSTITELI DATABÁZE	52
6.4.	POPIS INSTALACE KLIENTŮ ORACLE DATABASE	52
6.5.	DOKONFIGUROVÁNÍ INSTALACÍ	53
6.6.	POPIS PROVÁDĚNÍ TESTŮ	53
6.7.	VÝSLEDKY TESTŮ	53
7.	ZÁVĚR	54
	SEZNAM POUŽITÉ LITERATURY	55
	PŘÍLOHY	56
A-	SEZNAM PARAMETRŮ	56

SEZNAM OBRÁZKŮ

OBRÁZEK 1: AWR TIME MODEL STATISTICS REPORT	13
OBRÁZEK 2: DETAIL ARCHITEKTURY KOMUNIKACE ORACLE DATABASE-KLIENT/SERVER:.....	16
OBRÁZEK 3: ARCHITEKTURA KOMUNIKACE POUŽITÉ PRO APLIKACI TYPU KLIENT SERVER PRO ORACLE DATABASE S VYSVĚTLENÍM V NÁSLEDUJÍCÍM ODKAZU	17
OBRÁZEK 4: PŘIHLAŠOVACÍ FORMULÁŘ	37
OBRÁZEK 5: STAVOVÝ ŘÁDEK VE SPODNÍ ČÁSTI HLAVNÍHO FORMULÁŘE ZOBRAZUJÍCÍ STAV PŘIPOJENÍ A EDICI A VERZI DATABÁZE	39
OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA	41
OBRÁZEK 7:EXPORT.....	41
OBRÁZEK 8: PROFILER.....	47
OBRÁZEK 9: SCHÉMA TESTOVACÍHO PROSTŘEDÍ	51

1. ÚVOD

Lidé mají potřebu evidovat a shromažďovat informace už od pradávna. Tyto potřeby byly důvodem vzniku databáze, tedy jednoduše řečeno jakéhosi souboru informací. Dávno před vznikem databází lidé využívali papírové kartotéky, které umožňovaly zapsání, zatřídění a následné vyhledání potřebných informací, přičemž veškeré operace prováděl člověk. Postupem času s nástupem počítačů a následným propojování světa vznikaly potřeby uchovávat stále větší množství dat, které vyústily ve vývoj databázových technologií, které se již v polovině minulého století staly neodmyslitelnou součástí technologií informačních. Protože databáze slouží pro ukládání informací reálného světa, stala se součástí běžného života moderního člověka. Setkáme se s ní na každém kroku - ve školství jako evidence škol, zaměstnanců a žáků, ve zdravotnictví, v dopravě.. Jak již bylo zmíněno výše, primární potřebou je shromažďování dat a manipulace s nimi. Technologie pro přenos obsahu databázového systému do speciálních souborů mimo databázový systém se nazývá exportování, naopak technologie pro přenos dat z speciálních datových souborů uložených mimo databázový systém do databázového systému je importování.

Existuje velké množství firem zabývajících se databázovými technologiemi a jednou z předních firem specializovaných na databáze je firma **Oracle**. Ta pro účely přenosu dat vyvinula speciální nástroje, souhrnně nazývané **Oracle Database Utilities**. Některými z těchto nástrojů jsou originální **Export**, originální **Import** a **SQL*Loader**¹. Nástroje nabízejí velké množství parametrů pro nastavení požadavků na přenos dat. V základní edici **Oracle Database 10g ExpressEdition** jsou však tyto parametry v celém svém rozsahu dostupné pouze jako součást příkazů spouštěných z příkazové řádky. Edice **Oracle Database 10g ExpressEdition** je poskytována zdarma a nedisponuje grafickým rozhraním pro práci s těmito nástroji. Z tohoto důvodu je cílem této bakalářské práce naprogramování aplikace nazvané **OracleVisualTools**, umožňující uživateli pohodlnější užívání těchto nástrojů v uživatelsky příjemnějším grafickém rozhraní než je příkazová řádka. Uživatel bude mít jednoduše přehled o všech parametrech a jejich hodnotách a bude moci tyto hodnoty snadno modifikovat pomocí standardních ovládacích prvků. Aplikace navíc uživateli nabídne pohled do databáze na seznam objektů, ze kterých bude moci jednoduchým způsobem vybrat ty, které bude chtít exportovat/importovat. Uživatel tedy nebude muset znát obsah databáze a složitě přepisovat názvy objektů, se kterými bude chtít manipulovat. Další potřebou uživatele může být vytvoření zálohy dat. To se většinou děje pravidelně a často se stejnými parametry. Aplikace tedy nabídne uživateli možnost tyto parametry uložit a později provést export/import dat se stejným nastavením. Bude stačit jednoduše vybrat uložený seznam hodnot parametrů a tímto je jednoduše nastavit bez nutnosti nového zadávání hodnot parametrů. Bude však ponechána možnost editace načtených hodnot parametrů. Aplikace rovněž nabídne sledování průběhu procesu nástroje.

¹ Dále jen nástroje.

Dalším tématem této práce je řešení požadavků souvisejících s efektivitou práce s databází. Jednou z potřeb uživatelů jsou výkonnostní a s tím související časové nároky na manipulaci s daty v databázi. Pro zprostředkování dat z databáze nebo pro manipulaci s jejím obsahem byl vytvořen dotazovací jazyk SQL (Structured Query Language). Nesprávná formulace SQL příkazů v databázové aplikaci způsobuje většinu problémů s výkonem v jinak správně navržené a nakonfigurované databázi. Pro odstranění těchto problémů a zvýšení efektivity práce s databází byly vyvinuty nástroje shromažďující informace o vnitřní činnosti databáze z pohledu efektivity vykonávaných SQL příkazů. Většina nástrojů je ale zaměřena na zachycení historie provedených SQL příkazů. Cílem práce je tedy také online monitorování SQL příkazů, zobrazování statistik příslušných k těmto příkazům a možnost uložení a následného prohlížení historie provedených monitorování.

2. ANALÝZA

Tato kapitola poskytuje seznámení se základními pojmy týkajícími se databáze se zaměřením na nástroje **Oracle (Oracle Database Utilities)**, které jsou součástí práce, tedy **Export, Import** a **SQL*Loader**. Dále čtenáře uvede do problematiky zvyšování efektivity práce s databází, konkrétně pak on-line monitorování SQL příkazů spolu se shromažďováním jejich statistik. Nejdříve se však seznámíme s pojmem databáze.

„Databáze (neboli datová základna) je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i software (SŘBD).“[1]

Oracle Database je jedním ze systému řízení báze dat (Database Management System – DBMS), moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. **Oracle Corporation** je název firmy, oficiální název databázové platformy je **Oracle Database**.

Produkt **Oracle Database** je dodáván na trh v několika funkčně i výkonově odlišných edicích a též odlišně licencovaných:

Oracle Express Edition

Oracle Standard Edition One

Oracle Standard Edition

Oracle Enterprise Edition

Pouze **Oracle Database Express Edition** je volně šiřitelnou verzí. Ostatní edice jsou licencovány zpravidla podle počtu současně přístupujících uživatelů a podle počtu CPU. **“Oracle Database XE** je k dispozici pro 32bitové operační systémy **Linux** a **Windows**. Využije maximálně 1 CPU a 1 GB RAM serveru a umožňuje uchovávat až čtyři gigabajty uživatelských dat. Na serveru lze současně provozovat pouze jednu instanci **Oracle Database XE**.“ [2]

Další dělení produktu **Oracle Database** je dle čísla verze. V současnosti jsou podporovány verze 9i, 10g a 11g. Písmeno za číslem verze symbolicky vyjadřuje, na kterou oblast se Oracle zaměřil v dané verzi nejvíce (Oracle 9internet, Oracle 10grid a nyní Oracle 11growing the grid).

2.1. PŘEHLED APLIKACÍ ZABÝVAJÍCÍCH SE TEMATIKOU MANIPULACE S DATY A MONITOROVÁNÍM SQL PŘÍKAZŮ

Na obrázku přehled funkčních možností jednotlivých edicí Oracle Database:

Key Feature Summary	Express Edition 10g	Standard Edition One	Standard Edition	Enterprise Edition
Maximum	1 CPU	2 Sockets	4 Sockets	No Limit
RAM	1GB	OS Max	OS Max	OS Max
Database Size	4GB	No Limit	No Limit	No Limit
Windows	•	•	•	•
Linux	•	•	•	•
Unix		•	•	•
64 Bit Support		•	•	•
Application Development Read More				
Oracle SQL Developer Graphical tool that enhances productivity and simplifies database development tasks. Read More	•	•	•	•
Manageability Read More				
Enterprise Manager Manage and monitor all applications and systems based on the Oracle stack via a single, integrated console. Read More		•	•	•

OBRÁZEK I: PŘEHLED FUNKČNÍCH MOŽNOSTÍ JEDNOTLIVÝCH EDICÍ ORACLE DATABASE

[HTTP://WWW.ORACLE.COM/DATABASE/PRODUCT_EDITIONS.HTML](http://www.oracle.com/database/product_editions.html)

Z tabulky vyplývá, že všechny verze podporují volně šiřitelný produkt **Oracle SQL Developer**. **Oracle SQL Developer** je grafické rozhraní pro aplikaci **Oracle SQL*Plus**. Je to **Oracle** plně podporovaný grafický nástroj pro práci s databází, umožňuje pohodlným způsobem provádět základní úkoly, jako je procházení, vytváření, editování, mazání databázových objektů, lze spouštět SQL příkazy a SQL skripty, ladit PL/SQL kód, manipulovat s daty, exportovat/importovat data, vytvářet reporty. Umožňuje exportovat data tabulek do souboru v následujících formátech: XML (XML tagy a data), CSV (čárkou oddělené hodnoty, obsahuje hlavičku pro názvy sloupců), SQL Insert (INSERT příkazy), nebo **SQL*Loader** (SQL*Loader control file). Tento nástroj však nepodporuje on-line monitorování SQL příkazů a u nástrojů pro manipulaci s daty nenabízí všechny parametry dostupné z příkazové řádky.[3]

Kromě **Oracle Database10g Express Edition** disponují ostatní verze komplexním nástrojem **Oracle Enterprise Manager** - nástrojem pro správu a monitorování databází. Tento nástroj nabízí monitorování databáze pomocí AWR (Automatic Workload Repository- automatického úložiště dat) a pomocí nástroje **Automatic Database Diagnostic Monitor(ADDM)** server generuje doporučení, jak zlepšit výkon databáze. Od verze 10g obsahuje nástroj pro monitorování SQL příkazů **SQL Tuning Advisor**, od verze 11g **Automatic SQL Tuning** a **Real Time SQL Monitoring**. **RealTimeSQL Monitoring** automaticky monitoruje spuštěné SQL příkazy. Zobrazuje však pouze takové, které spotřebovávají více než 5s z času CPU nebo V/V času. Nezachytí tedy všechny spuštěné SQL příkazy. Pro používání **Oracle Enterprise Manager** a jeho rozšíření je potřeba zakoupit licenci.

Následující kapitoly nabídnou informace o možnostech připojení k databázi, originálních nástrojích **Export**, **Import** a **SQL*Loader** a možnosti monitorování databáze a konkrétně SQL příkazů spolu se shromažďováním statistik.

Tyto kapitoly jsou tematicky rozděleny podle oblasti, kterou se zabývají. První kapitola popisuje nástroje pro manipulaci s daty a objekty, tedy **Export**, **Import** a **SQL*Loader** a druhá kapitola **Profiler** se zabývá online monitorováním SQL příkazů a jejich statistikami.

2.2. Export, Import, SQL*Loader

Export, **Import** a **SQL*Loader**, jak již bylo zmíněno výše, jsou originální nástroje **Oracle**. Nástroje **Export** a **Import** slouží k zálohování a obnově databáze nebo k přenosu objektů mezi dvěma databázemi. Tyto databáze mohou běžet na různých platformách s rozdílnými konfiguracemi. **SQL*Loader** je nástroj pro přenos dat z externích souborů do databáze **Oracle**.

Export vykonává exportování objektů z databáze. Po vyexportování se tyto objekty uloží do binárního souboru s příponou .dmp, odkud mohou být následně importovány do jakékoli **Oracle** databáze. Verze **Importu** však musí být použita stejná nebo vyšší, než byla použita verze **Export** pro vznik tohoto binárního souboru.

Při **Exportu**, **Importu** a **SQL*Loaderu** je mnoho dostupných parametrů pro nastavení a které objekty mají být exportovány/importovány. Tyto parametry jsou popsány v příloze A- Seznam parametrů pro **Export**, **Import** a **SQL*Loader** a jsou pro všechny verze databáze/klienta stejné.

Od verze **Oracle Database 10g** jsou navíc dostupné nástroje **ExportDataPump** a **ImportDataPump**. Tyto nástroje nabízejí vylepšení v oblasti výkonu. Více však popisovány nebudou neboť nejsou zadáním práce.

Možnost použití nástrojů **Export**, **Import** a **SQL*Loader** je zajištěna spustitelnými soubory exp.exe, imp.exe a sqlldr.exe. Tyto soubory jsou součástí instalace **Oracle Database Serveru**. Aby byly současně součástí klienta, musí uživatel při instalaci **Oracle Database Clienta** nastavit administrátorský mód. Jsou poté dostupné v domovském adresáři Oracle produktu %ORACLE_HOME%\bin\. Jelikož je typicky **Oracle Database Server** nainstalován na jiném stroji než **Oracle Database Client**, bude tedy po uživateli na straně klienta pro funkčnost nástrojů Export, Import a SQL*Loader vyžadována existence těchto nástrojů na stanici.

V **Oracle Database Express Edition** jsou originální nástroje **Export**, **Import** a **SQL*Loader** spustitelné pouze z příkazové řádky, jak již bylo uvedeno výše. Spouštění těchto příkazů z příkazové řádky může být realizováno třemi různými způsoby. Příklady jsou uváděny pro nástroj Export (klíčové slovo exp může být doplněno příponou .exe), pro nástroje Import stačí nahradit exp výrazem imp, příp. imp.exe, pro SQL*Loader řetězcem sqlldr, příp. sqlldr.exe.

- ☛ Vložením názvu příkazu, uživatelského jména, hesla a parametrů na příkazovou řádku ve formátu

```
exp username/password PARAMETER=value
exp username/password
PARAMETER=(value1,value2,...,valuen)
```

- Interaktivně pouze zadáním exp(exp.exe)a uživatelského jména a hesla, uživatel je pak dotazován na jednotlivé hodnoty parametrů, nejsou mu však nabídnuty všechny

```
exp username/password
```

- Pomocí parametrického souboru s příponou .par obsahujícím parametry pro Export. Parametrický soubor může být následován parametry, které mohou být shodné jako parametry v souboru, při výběru ze dvou hodnot jednoho parametru se použijí parametry zadané na příkazové řádce. Z důvodu bezpečnosti se nedoporučuje do parametrického souboru uvádět username/password , přestože je to možné.

```
exp PARFILE=filename
```

```
exp username/password PARFILE=filename
```

možnosti zadávání hodnot do parametrického souboru:

```
PARAMETER=value
```

```
PARAMETER=(value)
```

```
PARAMETER=(value1, value2, ...)
```

Cesty k nainstalovanému software **Oracle** se ukládají do proměnné prostředí PATH. Po zadání samotného příkazu a parametrů **Exportu (Importu, SQL*Loaderu)** na příkazovou řádku se použije první soubor exp.exe(imp.exe,sqlldr.exe), který bude nalezen v proměnné prostředí PATH v adresáři obsahujícím nainstalovaný software **Oracle**. Nastavování této proměnné tak, aby se vždy adresář se správnou verzí vyskytoval před všemi ostatními adresáři s nainstalovanými produkty Oracle je nepohodlné a nevýhodné z toho důvodu, že ji může kdykoliv kdokoliv změnit.

Aby se tedy předešlo nejistotě, jaká verze **Exportu(Importu, SQL*Loaderu)** se vlastně spustí a aby se předešlo nutnosti nastavování cesty v proměnné prostředí PATH, stačí se v příkazové řádce nastavit do cesty proměnné ORACLE_HOME příslušné verze serveru nebo klienta. To zajistí spuštění správné verze. Tato cesta ze zjistí pomocí dat v registru operačního systému. Zde se totiž po instalaci **Oracle Database Server** nebo **Oracle Database Client** verze 9i, 10g nebo 11g vytvoří v adresáři LOCALMACHINE\SOFTWARE\ORACLE\%Folder% proměnná ORACLE_HOME obsahující cestu k adresáři bin příslušného Oracle produktu.

Tato poslední možnost zajištění výběru požadované verze nástroje byla vybrána jako vhodné řešení, neboť lze z registru zjistit potřebné cesty k souborům exp.exe, imp.exe a sqlldr.exe daném stroji procházením všech podsložek složky LOCALMACHINE\SOFTWARE\ORACLE\ a hledáním hodnot proměnných ORACLE HOME. Tyto hodnoty pak mohou být uživateli nabídnuty pro jednoduchý výběr verze nástroje.

2.3. PROFILER

2.3.1. Monitorování a ladění databáze obecně

Pro databázové administrátory a vývojáře aplikací je ladění výkonu aplikace velmi důležitou oblastí, je součástí životního cyklu každé databázové aplikace. Zaměřuje se na nalezení a odstranění problémů, které vedou k nízkému výkonu. Začíná analýzou toho, čím se databázová aplikace zabývá a jaké činnosti spotřebovávají nejvíce zdrojů. Poté se volí co se bude ladit:

- ▣ příkazy jazyka SQL
- ▣ využití paměti
- ▣ využití úložiště dat
- ▣ příkazy pro manipulaci s daty
- ▣ fyzické a logické úložiště
- ▣ síťový provoz

Klíčem pro ladění příkazů jazyka SQL je minimalizovat cesty získávání dat, které aplikace využívá při vyhledávání dat. I jediný dotaz, který není dostatečně dobře vyladěný, může mít vliv na výkon celé databáze v případě, že je prováděn velmi často. Z tohoto důvodu jej cílem této práce vytvoření nástroje, který by umožňoval online monitorování SQL příkazů aktuálně spuštěných v databázi **Oracle** spolu se zobrazováním statistik a rovněž nabídnout uživateli možnost procházet historii monitorování. Pro krátký úvod k pochopení, co se vlastně v databázi děje po spuštění SQL příkazu slouží následující kapitola.

Technologie zpracování SQL příkazu

Poté, co uživatel spustí SQL příkaz, provede se několik kroků. Prvním krokem je tzv. parsování, kdy se provádí syntaktická a sémantická analýza dotazu. Toto parsování je dvojího typu: soft a hard. Soft se použije v případě, kdy byl již SQL příkaz parsován a je dostupný v paměti. Naopak v případě typu hard musí být vykonány všechny kroky parsování. Soft parsování tedy zlepšuje výkon systému, zatímco hard parsování výkonost systému snižuje.

Poté přichází na řadu optimalizátor, který má za úkol navrhnout co nejefektivnější způsob získání výsledku dotazu. Využije jednu ze dvou metod: cost-based (CBO) nebo rule-based(RBO). Cost Based se použije má-li optimalizátor k dispozici interní statistiky, rule-based v opačném případě. Poté se vytvoří optimalizační plán, který následně zpracuje Row Source Generator. Ten vytvoří exekuční plán pro daný SQL dotaz. Exekuční plán má formu stromu, jehož uzly jsou tvořeny možnými přístupovými cestami(index scan, sequential scan), spolu s algoritmy(hash join, nested loops). Vykonáváním jednotlivých řádků exekučního plánu se získá výsledek zadaného SQL příkazu.

Následné řádky přiblíží, jak vlastně samotný proces vytváření a používání statistik probíhá a jaké nástroje pro monitorování používá *Oracle*.

Oracle a statistiky

Pro účinné diagnostikování výkonu systému musí být k dispozici statistiky. Databázové statistiky poskytují informace o typu zátěže databáze a interních a externích zdrojích užívaných databází **Oracle**.

Základem shromažďování statistik je Automatic Workload Repository (AWR). AWR je zdrojem informací pro několik Oracle 10g nástrojů jako jsou např. **Automatic Database Diagnostic Monitor** a **SQL Tuning Advisor**. Pracuje tak, že po spuštění databáze začne automaticky snímkovat celou databázi pravidelně jednou za hodinu a tyto snímky se uchovávají po dobu 7 dní (při výchozím nastavení). Snímkování databáze je vykonáváno na pozadí pomocí procesu MMON. Tento proces je přímo určen k vytváření snímků databáze a k jejich ukládání do tabulkového prostoru SYSAUX. AWR snímky uchovávají hodnoty čítačů v čase vytvoření snímku. Poté, co se vytvoří snímek, uloží se hodnoty čítačů. Samotná hodnota čítače je nic neříkající. Důležitá se stává ve chvíli, kdy je provedeno více snímkování a hodnoty jednotlivých čítačů se mohou porovnávat. Tímto způsobem vznikají analýzy. Například je tedy možné identifikovat SQL příkazy, které spotřebovaly velké množství CPU času, či provedly mnoho operací čtení s disku během určitého časového intervalu. Nesmí však dojít k restartu databáze, kdy jsou čítače vynulovány. Srovnávací reporty funkce AWR pak poskytují cenné informace pro rychlé řešení výkonových problémů. [4],[5]

Pro představu uvedeme příkaz pro vytvoření snímku:

```
SQL> exec dbms_workload_repository.create_snapshot;
```

AWR se používá ke shromažďování statistik, tyto statistiky se dělí podle typu zátěže databáze. Nejdůležitějšími statistikami jsou:

- Wait events
- Time-model statistiky ve v \$SESS_TIME_MODEL a V\$SYS_TIME_MODEL pohledech
- Active Session History (ASH) statistiky v pohledu V\$ACTIVE_SESSION_HISTORY
- Systémové statistiky a statistiky sessions ve V\$SYSSTAT and V\$SESSTAT pohledech
- SQL příkazy náročné na zdroje

Wait events jsou statistiky které indikují, že session musí čekat na nějakou událost než bude moci pokračovat ve vykonávání své činnosti. Zatímco session čeká, AWR zaznamenává konkrétní Wait event, přičemž existuje více než 800 různých Wait events a každá z nich patří do jedné ze 12 tříd:

- Administrative: Wait events, které jsou způsobeny DBA příkazy, př.: index rebuild.
- Application: čekání způsobuje kód aplikace uživatele, př: row level lock.
- Cluster: čekání v souvislosti s Real Application Cluster, př: gc cr block busy.
- Commit: př: log file sync.
- Concurrency: čekání na interní zdroje databáze, př: Latch wait

- Configuration: čekání způsobené neadekvátní konfigurací databáze nebo zdrojů instance, př: Undersized log file size, undersized shared pool size
- Idle: čekání které zaznamenává, že session je neaktivní, tato session čeká na práci, př: SQL*Net message from client.
- Network: čekání vztahující se ke zprávám zaslaným po síti, př: SQL*Net more data to dblink.
- Other: čekání, které se typicky nevyskytují na systému, př: wait for EMON to spawn.
- Scheduler: čekání spojené s Resource Managerem, př: resmgr: become active
- System I/O: čekání na V/V procesy na pozadí, př: DBWR waits for 'db file parallel write'.
- User I/O: čekání na uživatele V/V, př: 'db file sequential read'.

TIMEMODEL statistiky

Obsahují informace, kde je stráven DB čas během jednoho intervalu mezi snímky.

```

Time Model Statistics DB/Inst: LSQ/lsq Snaps: 1355-1356
-> ordered by Time (seconds) desc

```

Statistic Name	Time (seconds)	% Total DB Time
DB time	7,274.60	100.00
sql execute elapsed time	7,249.77	99.66
background elapsed time	778.48	10.70
DB CPU	150.62	2.07
parse time elapsed	45.52	.63
hard parse elapsed time	44.65	.61
PL/SQL execution elapsed time	13.73	.19
background cpu time	8.90	.12
PL/SQL compilation elapsed time	3.80	.05
connection management call elapsed time	.15	.00
Java execution elapsed time	.05	.00
hard parse (bind mismatch) elapsed time	.00	.00
hard parse (sharing criteria) elapsed time	.00	.00
sequence load elapsed time	.00	.00
failed parse (out of shared memory) elapsed	.00	.00
inbound PL/SQL rpc elapsed time	.00	.00
failed parse elapsed time	.00	.00

OBRÁZEK 1: AWR TIME MODEL STATISTICS REPORT

http://www.dba-oracle.com/oracle10g_tuning/t_time_model_statistics.htm

ActiveSessionHistory(ASH) statistiky obsahují snímky aktivních sessions a jsou ukládány do SystemGlobalArea(SGA). Každá session, která je připojena do databáze a spotřebovává CPU time nebo čeká na nějakou událost s výjimkou událostí tříd IdleClass je aktivní session. Snímky jsou vytvářené každou sekundu v gv_\$active_session_history, zde jsou uloženy průměrně po dobu 30 minut a poté jsou uloženy do dba_hist_active_sess_history.

Nástroje Oracle pro monitorování databáze

Zřejmě nejvíce propracovaným nástrojem pro správu a monitorování databáze je **OracleEnterpriseManager(OEM)** firmy **Oracle**.

Oracle Enterprise Manager používá pro zpracování a zobrazení informací o výkonnosti data pomocí výše popsaného Automatic Workload Repository. Jednou z metod pro diagnostiku problémů spojených s výkonem je *Automatic Database*

Diagnostic Monitor (ADDM). **ADDM** se spustí poté, co jsou sesbírány potřebné snímky a tyto snímky následně použije pro analýzy aktivity databáze. Výsledkem tohoto nástroje je pak poskytnutí doporučení k ladění. Následující seznam obsahuje některé typy problémů, kterými se **ADDM** zabývá:

- spotřebou času CPU- zde je možno odhalit, zda čas CPU spotřebovává **Oracle** nebo jiná aplikace
- velikostí struktur- zda jsou paměťové struktury jako PGA, SGA a buffer cache adekvátně velké
- SQL příkazy náročnými na zdroje systému
- vysokým zatížením zpracování PL/SQL kódu

OEM dále nabízí složku *Historical SQL*. Tato složka umožňuje zobrazení historie SQL příkazů uložených v AWR po 24 hodin. V grafech a tabulkách pak zobrazuje analýzy výkonu a spotřeby zdrojů SQL příkazů. Na tyto problematické SQL příkazy se poté může spustit **SQL Tuning Advisor** pro získání doporučení zlepšení výkonu.

Další zajímavou složkou je *Top Consumers*. Zabývá se konzumenty spotřebovávajícími nejvíce zdrojů. Zobrazuje 10 těchto největších konzumentů podle session, module, action nebo client identifikátoru. Umožňuje také povolit sbírání statistik a trasování SQL s následným pohledem do trace souboru. [6]

Dalším nástrojem pro ladění je balíček **Statspack**. Je to soubor SQL a PL/SQL kódu, který dohromady umožňuje snímkování stavu databáze v definovaných intervalech, uložení těchto informací a jejich následné vyhodnocení. Tento nástroj byl však ve verzi **Oracle Database 10g** nahrazen **ADDM**. **Statspak** však svou činností zatěžuje server a snímkováním nelze zachytit všechny SQL příkazy.

Oracle však i v edici **ExpressEdition** nabízí balíček **DBMS_MONITOR**. Tento nástroj obsahuje procedury pro zahájení a ukončení trasování SQL příkazů a shromažďování jejich statistik. Dokáže zachytit všechny vykonávané SQL příkazy vykonávané v rámci zadané session, client identifikátoru nebo module a action name. Výstupem balíčku **dbms_monitor** je trace soubor uložený v cestě, kde je nainstalován server databáze v adresáři \udump.

2.4. Diskuze k řešení Profileru

Trasování SQL příkazů přímo do tabulky v databázi by umožňovalo získávání dalších statistik pomocí propojování pohledů. Tato metoda by však mohla ovlivnit výkon databáze a zkreslit výsledky. Pokud bychom znali SQL příkaz, stačilo by před zadáním zadat start monitoring, provést příkaz, provést stop monitoring a pomocí pohledů v\$statname a v\$sesstat odečíst hodnoty statistiky před a po vykonání dotazu a tím získat nové statistiky.

Získání statistik by se dalo realizovat následujícím dotazem:

```
SELECT ses.username,n.name,stat.*
FROM v$session ses, v$sesstat stat, v$statname n
WHERE ses.sid=stat.sid and stat.statistic#=n.statistic#;
```

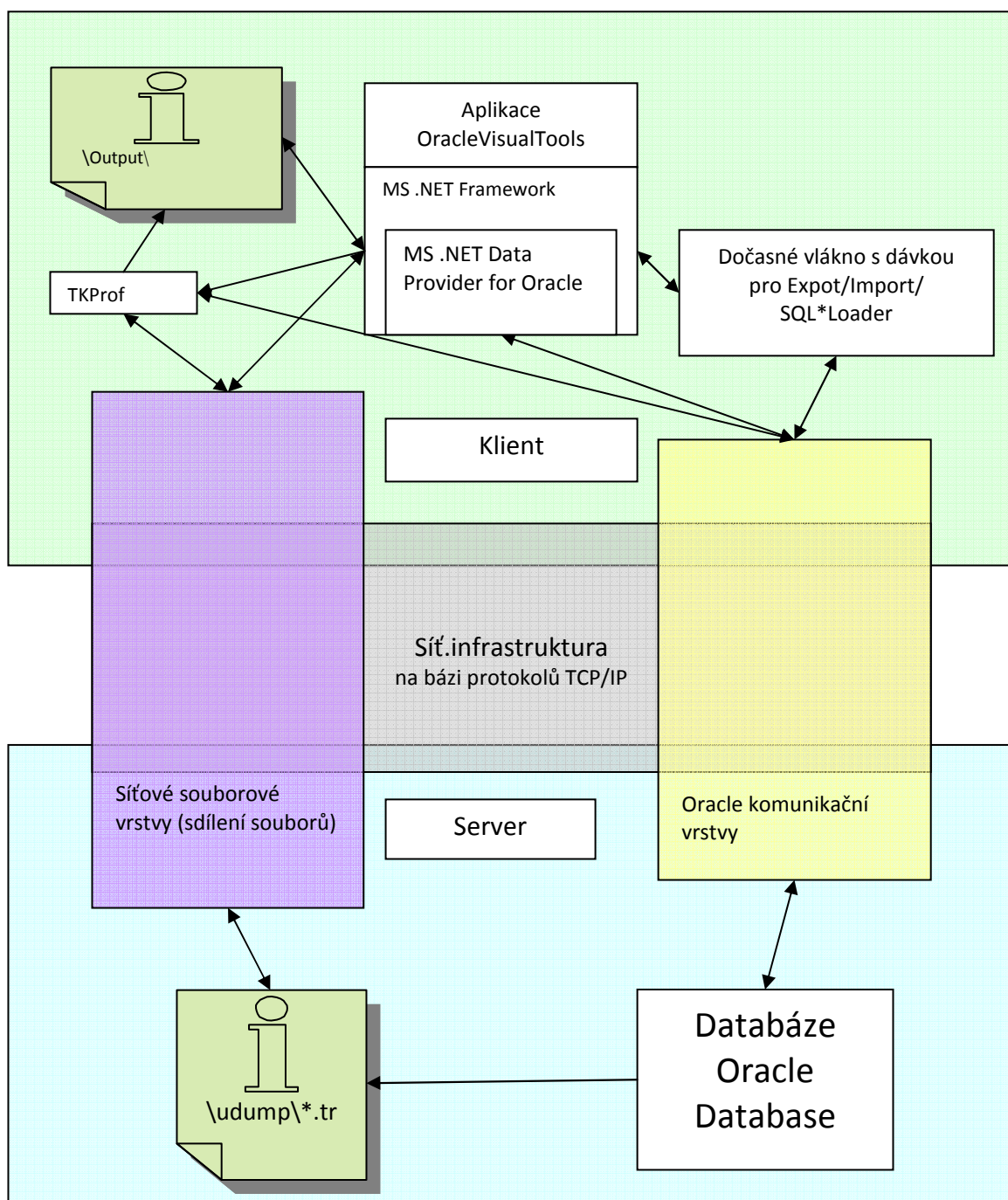
Prvním problémem monitorování v *Oracle Database Express Edition* je tedy získat veškeré SQL příkazy uživatelů, neboť neobsahuje žádné tabulky s historií všech SQL dotazů.

První myšlenkou k získání SQL příkazů bylo propojit pohledy `v$session` a `v$sql` následujícím způsobem:

```
select u.sid, u.username s.sql_text from v$sql s, v$session u
        WHERE s.hash_value = u.sql_hash_value order by u.sid;
```

Tímto lze však získat pouze aktuálně prováděný SQL příkaz. V úvahu tedy připadalo nad tímto dotazem vytvořit JOB a nastavit sekvence provádění na zadaný časový interval. Pokud bychom se spokojili s tím, že SQL příkazy, které se vykonávají kratší dobu než je specifikovaný časový interval pro nás nejsou podstatné, pak by se o tomto řešení dalo uvažovat, celkově jsem ho ale vyhodnotila za nevyhovující. Úlohu neřeší, neboť úkolem aplikace **OracleVisualTools** je zachytit všechny SQL příkazy. Využití by se spíše našlo ve chvíli, kdy administrátor přijde k nové databázi a chce si udělat obrázek co se v databázi děje.

Nejlepším řešením se tedy ukázalo být použití balíku **DBMS_MONITOR**. Je dostupný i ve verzi ExpressEdition a jeho výhodou je zachycení všech SQL příkazů vykonaných v databázi a shromažďování základních statistik, pomocí nichž se dají odhalit problematická místa snižující výkon. Monitorovat se dá celá databáze, vybrané sessions, vybraný client_identifier a nebo vybrané module a action name.



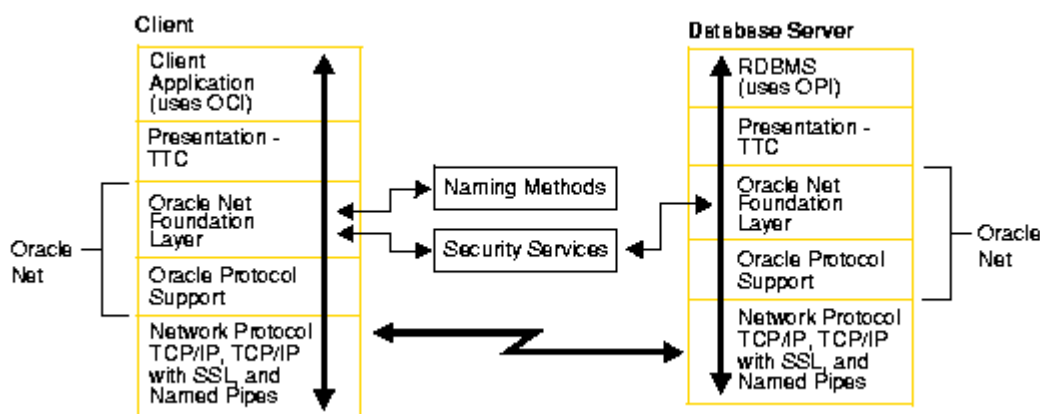
OBRÁZEK 2: DETAIL ARCHITEKTURY KOMUNIKACE ORACLE DATABASE-KLIENT/SERVER:

3. DESIGN ŘEŠENÍ

Celá úloha je rozdělena na tři části. První část se zabývá architekturou komunikace **OracleVisualTools**, druhá nástroji **Export**, **Import** a **SQL*Loader** a závěrečnou třetí částí je řešení nástroje **Profiler**.

3.1. PŘEHLED ARCHITEKTURY KOMUNIKACE ORACLEVISUALTOOLS

3.1.1. Architektura komunikace OracleVisualTools – databáze a souborový systém



OBRÁZEK 3: ARCHITEKTURA KOMUNIKACE POUŽITÉ PRO APLIKACI TYPU KLIENT SERVER PRO ORACLE DATABASE S
VYSVĚTLENÍM V NÁSLEDUJÍCÍM ODKAZU

http://download.oracle.com/docs/cd/B10501_01/network.920/a96580/architec.htm

Připojovací řetězec

V adresáři ORACLE HOME\NETWORK\ADMIN se může nalézat soubor tnsnames.ora. Obvykle se vytváří při instalaci **Oracle Database Serveru** a **Oracle Database Clientu**, není to však podmínkou. Pokud se soubor tnsnames.ora nevytvořil, může si jej uživatel na uvedeném místě vytvořit sám. V tomto souboru jsou definována připojení spolu s aliasy ve formátu:

```
Network_alias =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = port))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = service_name)  
    )  
  )
```

Pokud bude tento soubor nalezen, bude uživateli nabídnuta možnost zadat pouze Network_alias a parametry host, port a service_name mohou být ze souboru vyfiltrovány a uživateli zobrazeny.

Připojení k databázi umožňuje knihovna .NET Framework Data Provider for Oracle. Syntaxe connection stringu je následující:

```
User Id = ; Password=;
DataSource=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port
))
(CONNECT_DATA=(SERVICE_NAME=service_name))); "
```

[7]

Pokud uživatel přímo zadá parametry `host`, `port` a `service_name`, budou do tohoto řetězce vloženy přímo, pokud zadá jako ty připojení TNS, bude vyhledán soubor `tnsnames.ora`, z něj tyto hodnoty vyfiltrovány a vloženy do řetězce.

3.1.2. Architektura komunikace aplikace **OracleVisualTools** využívající prostředí **Microsoft .Net Framework** pro připojení a práci s **Oracle Database** a pro realizaci **Profileru**

Strana Klienta

Aplikace **OracleVisualTools** – realizace připojení a komunikace s databází *Microsoft .NetFramework Data Provider for Oracle* (obsahuje objekty *Connection*, *Command*, *DataReader*, *DataAdapter*, *DataSet*) – součást prostředí *Microsoft .Net Framework*

OCI Oracle Call Interface

Presentation - TTC

Oracle Net Foundation Layer

Oracle Protocol Support

Network Protocol (TCP/IP, TCP/IP s SSL, a Named Pipes)

Strana Serveru

Network Protocol (TCP/IP, TCP/IP s SSL, a Named Pipes)

Oracle Protocol Support

Oracle Net Foundation Layer

Presentation - TTC

RDBMS(používá OPI)

3.2. EXPORT, IMPORT, SQL*LOADER

Po nainstalování **Oracle Database Server** a **Oracle Database Client** jsou nástroje originální **Export**, **Import** a **SQL*Loader** dostupné ve formě souborů `exp.exe`, `imp.exe` a `sqlldr.exe` v adresáři `ORACLE_HOME\bin`. Spouští se z příkazové řádky následujícím způsobem:

```
exp username/password@//host:port/service_name
exp username/password@TNS:network alias
```

kdy parametry `host`, `port` a `service_name` nebo `network_alias` byly zadány při připojení.

Pro Oracle 9i nefunguje připojení pomocí `host`, `port` a `service_name`, proto bude nutné připojovat se pouze pomocí aliasu.

Soubory exp.exe, imp.exe a sqlldr.exe jsou dostupné na straně serveru a pokud se instaloval klient v administrátorském módu, jsou součástí i klienta. Přestože jsou spustitelné na příkazové řádce, není vhodné parametry a jejich hodnoty vkládat přímo na jednu řádku za typ nástroje a přípojovací řetězec z důvodu omezení délky příkazové řádky. Vhodným řešením je naplnit parametry z grafického rozhraní do parametrického souboru, vytvořit batch soubor do kterého se na začátek vloží řetězec pro nastavení se v příkazovém řádku do ORACLE HOME\bin. Obsah souboru tedy bude pro ORACLE HOME C:\oracle\app\oracle\product\10.2.0\server a připojení pomocí host, port a service_name vypadat následovně:

C:

\oracle\app\oracle\product\10.2.0\server\bin

Exp.exe username/password@//host:port/service_name

parfile=uziv_parfile.txt

3.2.1. Architektura komunikace při realizaci vlákna obsahující parametrizovanou dávku na spuštění příkazů exp.exe, imp.exe, sqlldr.exe, tkprof.exe

Strana Klienta

Aplikace **OracleVisualTools** – realizace vlákna s dávkou pro **Export, Import SQL*Loader** nebo **TKProf**

Stiskem tlačítka v grafickém prostředí pro spuštění nástroje uživatel odstartuje v aplikaci **OracleVisualTools** činnost pro vytvoření nového vlákna zajišťující zpracování příkazů exp.exe, imp.exe a sqlldr.exe s parametry předanými z grafického prostředí. Vlákno zabezpečuje provedení dávky, a logy, chybová hlášení a návratový kód jsou přes vlákno vráceny zpět do aplikace ke zpracování. Příkazy exp.exe, imp.exe nebo sqlldr.exe (z Oracle Utilities) a tkprof.exe samostatně komunikují v pořadí:

OCI Oracle Call Interface

Presentation - TTC

Oracle Net Foundation Layer

Oracle Protocol Support

Network Protocol (TCP/IP, TCP/IP with SSL, and Named Pipes)

Strana Serveru

Network Protocol (TCP/IP, TCP/IP with SSL, and Named Pipes)

Oracle Protocol Support

Oracle Net Foundation Layer

Presentation - TTC

RDBMS(používá OPI)

3.3. PROFILER

Jak jsem již zmínila výše, vybrala jsem jako způsob řešení balíček **dbms_monitor**. Umožňuje trasování podle sessions, client_id, module a action name nebo celé databáze.

Možností dosáhnoutí vytvoření trace souboru a shromažďování statistik je však celá řada:

```
SQL> ALTER SESSION SET sql_trace=TRUE;
SQL> ALTER SESSION SET sql_trace=FALSE;

SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => TRUE);
SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => FALSE);

SQL> ALTER SESSION SET EVENTS '10046 trace name context forever,
level 8';
SQL> ALTER SESSION SET EVENTS '10046 trace name context off';

SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123,
serial#=>1234, sql_trace=>TRUE);
SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123,
serial#=>1234, sql_trace=>FALSE);

SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046, le=>8,
nm=>' ');
SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046, le=>0,
nm=>' ');
[8]
```

Formát trace souboru by měl být stejný, ať se trasování spustí přes **dbms_monitor**, alter session set **SQL_TRACE=true**, nebo nastavení **EVENT=10046**... Pomocí **dbms_monitor** se však nemusí na začátku sledované session spouštět speciální příkaz pro zapnutí trasování přímo v dané session a trasování se da zapnout pro jinou session nebo client identifier, module a action name nebo ceou databázi.

Soustředila jsem se na monitorování jednotlivých sessions jako nejmenších celků oproti monitorování celé databáze.

Monitorování jedné session spustíme příkazem:

```
DBMS_MONITOR.SESSION_TRACE_ENABLE(
    session_id    IN    BINARY_INTEGER DEFAULT NULL,
    serial_num    IN    BINARY_INTEGER DEFAULT NULL,
    waits         IN    BOOLEAN DEFAULT TRUE,
    binds         IN    BOOLEAN DEFAULT FALSE)
```

Monitorování vypneme příkazem:

```
DBMS_MONITOR.SESSION_TRACE_DISABLE(
    session_id    IN    BINARY_INTEGER DEFAULT NULL,
    serial_num    IN    BINARY_INTEGER DEFAULT NULL);
```

Kde *session_id* je číslo session, kterou chceme monitorovat a *serial_num* specifikujeme pro případ, že by session s jedním id zanikla a poté zase vznikla. Pokud specifikujeme *serial_num* na null, pak je monitorována každá session se zadaným *session_id*.

Každý časový okamžik session je monitorován. Jestliže session pracuje, spotřebovává čas CPU, jestliže session čeká, spotřebovává wait time. Wait time se dělí na dva případy. Session buď na něco čeká, nebo je v nečinnosti(idle time). Když Oracle spustí SQL příkaz, tento příkaz někdy musí čekat na specifickou událost- wait event.

Pokud nastavíme parametr waits na true, nastavíme sledování wait events. Nastavením parametru binds na true umožníme sledovat i proměnné.

Dbms_monitor shromažďuje informace do trace filů, defaultně umístěných v adresáři \UDUMP. Cesta k těmto souborům se dá jednoduše zjistit pomocí pohledů *v\$session*, *v\$process*, *v\$instance* a *v\$parameter* SQL příkazem:

```
SELECT a.trace_path || '\\\ ' || d.instance_name || '_' ||  
b.trace_file  
FROM (SELECT VALUE trace_path FROM v$parameter  
WHERE NAME = 'user_dump_dest') a,  
SELECT instance_name FROM v$instance) d,  
SELECT      'ora_' || spid || (SELECT NVL2(VALUE, '_' || VALUE,  
NULL) AS VALUE FROM v$parameter  
WHERE NAME = 'tracefile_identifier') || '.trc' trace_file FROM  
v$process p, v$session s  
WHERE (p.addr = s.paddr AND s.sid=xxx AND s.serial#=yyy))b
```

Kdy cesta je uložena ve *v\$parameter* v proměnné 'user_dump_dest', název souboru sestává z názvu instance_ora_id procesu.

Trace soubor je vztažen ke konkrétní session. Pokud vypneme a znovu zapneme trace v rámci jedné session, Oracle bude zapisovat do toho samého souboru. Pokud ho mezitím smažeme, daná session už nezapiše nic.

Příklad obsahu trace souboru:

```
Dump file c:\oracle\app\oracle\admin\xe\udump\xe_ora_2704.trc  
Sun Aug 02 15:20:16 2009  
ORACLE V10.2.0.1.0 - Production vsnsta=0  
vsnsql=14 vsnxtr=3  
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production  
Windows XP Version V5.1 Service Pack 2  
CPU : 2 - type 586  
Process Affinity : 0x00000000  
Memory (Avail/Total): Ph:150M/1022M, Ph+PgF:1119M/2456M,  
VA:1606M/2047M  
Instance name: xe
```

Redo thread mounted by this instance: 1

Oracle process number: 20

Windows thread id: 2704, image: ORACLE.EXE (SHAD)

```
*** 2009-08-02 15:20:16.481
*** SERVICE NAME:(SYS$USERS) 2009-08-02 15:20:16.309
*** CLIENT ID:(eva:127.0.0.1) 2009-08-02 15:20:16.309
*** SESSION ID:(31.257) 2009-08-02 15:20:16.309
=====
PARSING IN CURSOR #2 len=37 dep=1 uid=0 oct=3 lid=0 tim=107681468090
hv=1398610540 ad='20ae9760'
select text from view$ where rowid=:1
END OF STMT
PARSE
#2:c=0,e=116477,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681468075
EXEC
#2:c=0,e=110245,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681798557
FETCH
#2:c=0,e=89,p=0,cr=2,cu=0,mis=0,r=1,dep=1,og=4,tim=107681803410
STAT #2 id=1 cnt=1 pid=0 pos=1 obj=63 op='TABLE ACCESS BY USER ROWID
VIEW$ (cr=1 pr=0 pw=0 time=71 us)'
=====
PARSING IN CURSOR #3 len=43 dep=0 uid=5 oct=3 lid=5 tim=107681847549
hv=1991533359 ad='20a90e1c'
select tablespace_name from dba_tablespace$
END OF STMT
PARSE
#3:c=93750,e=545107,p=0,cr=2,cu=0,mis=1,r=0,dep=0,og=1,tim=107681847
540
EXEC #3:c=0,e=74,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=107681869624
WAIT #3: nam='SQL*Net message to client' ela= 10 driver
id=1952673792 #bytes=1 p3=0 obj#=37 tim=107681894384
FETCH
#3:c=0,e=157,p=0,cr=7,cu=0,mis=0,r=5,dep=0,og=1,tim=107681899989
*** 2009-08-02 15:20:35.621
WAIT #3: nam='SQL*Net message from client' ela= 18880225 driver
id=1952673792 #bytes=1 p3=0 obj#=37 tim=107700785223
STAT #3 id=1 cnt=5 pid=0 pos=1 obj=16 op='TABLE ACCESS FULL TS$
(cr=7 pr=0 pw=0 time=138 us)'
=====
```

Popis obsahu trace souboru

V hlavičce se nachází základní informace jako je název trace souboru, datum vytvoření souboru, tedy první spuštění monitorování na dané session, verze Oracle database, jméno instance, service name, client id a session id.

Následuje SQL příkaz s přiděleným cursorem, a dalšími parametry:

- ▣ len = délka příkazu
- ▣ dep= hloubka příkazu- pokud příkaz nebyl rekursivní, má hloubku příkazu 0, jinak hodnotu větší než 0
- ▣ uid = id schematu, pod kterým bylo provedeno parsování SQL dotazu id schematu
- ▣ oct = typ příkazu: SELECT, INSERT, UPDATE, DELETE
- ▣ lid = id privilegovaného uživatele
- ▣ hv = hash id
- ▣ ad = adresa SQL příkazu pro dohledání v pohledu v\$sqlarea

Následuje parsování příkazu, provedení a získání dat. Statistiky k těmto akcím najdeme na řádcích začínající PARSE, EXECUTE a FETCH. Jejich statistiky jsou následující :

- ▣ `c` = čas procesoru
- ▣ `e` = uplynulá doba
- ▣ `p` = počet fyzických čtení
- ▣ `cr` = počet bufferů získaných pro CR čtení
- ▣ `cu` = počet bufferů získaných v aktuálním nastavení
- ▣ `mis` = kurzory ponechané v cache
- ▣ `r` = počet zpracovaných řádků
- ▣ `dep` = hloubka příkazu- pokud příkaz nebyl rekursivní, má hloubku 0, jinak >0
- ▣ `og` = cíl optimizéru: 1=All_Rows, 2=First_Rows, 3=Rule, 4=Choose
- ▣ `tim` = časová známka, která se dá se propojit s pohledem `v$timer`

Aktuální čas se pak dá z parametru `tim` zjistit z pohledu `v$timer`.

Statistiky STAT:

`<CURSOR>` = kurzor, ke kterému se vztahují statistiky

`id` = řádek vysvětluje plán dotazu. Toto je efektivita jednotlivých řádků v "execution tree"

`cnt` = počet řádků

`pid` = nadřazené id tohoto řádku

`pos` = pozice v "explain plan"

`obj` = id objektu tohoto řádku (pokud se jedná o základní objekt)

`op` = '...' přistupující operace k řádku

WAIT událost, na kterou se čeká

`nam` = jméno wait eventu

`ela` = uplynulý čas

`p1` = proměnná p1 pro wait event

`p2` = proměnná p2 pro wait even

`p3` = proměnná p3 pro wait event

BIND proměnné

`Value` = hodnota proměnné

Zmiňované pohledy:

`v$session`- pohled na aktuální sessions

`v$sql`- pohled obsahující statistiky nad shared SQL area. Shared SQL area je důležité pro vykonání každého SQL příkazu obsahující i exekuční plán.

`v$statname`- pohled na seznam statistik

`v$sesstat`- pohled na statistiky pro jednotlivé sessions

`v$parameter`- pohled na inicializační parametry

`v$instance`- pohled na stav současné instance

`v$process`- pohled na procesy

TKProf

Aplikace **OracleVisualTools** nabízí také využití nástroje TKProf. Tento nástroj transformuje vytvořené trace soubory do čitelnější podoby.

Tento nástroj je dostupný v adresáři ORACLE_HOME/bin souborem tkprof.exe a používá se následujícím způsobem:

```
ORACLE_HOME\bin tkprof.exe trace_soubor.trc output.txt
```

Nabízí větší množství parametrů, podle kterých je možné obsah trace souboru modifikovat například podle toho, kterou statistiku chceme sledovat. Seznam možných nastavení je k nahlédnutí zde:

<http://www.oracleutilities.com/OSUtil/tkprof.html>

Aplikace OracleVisualTools však zatím použití parametrů nenabízí.

4. PROGRAMÁTORSKÁ DOKUMENTACE

V následujícím textu je popsáno pomocí jakých nástrojů byla aplikace OracleVisualTools naprogramována a jak byla řešena z hlediska zdrojového kódu. Jsou zde především popsány nejdůležitější třídy, realizace připojení k databázi a získávání dat z databáze, spuštění balíčku **dbms_monitor** nezbytného pro nástroj Profiler a realizace spuštění **Exportu**, **Importu** a **SQL*Loaderu**.

V zadání bakalářské práce nebyly popsány omezení na použitý operační systém ani na vývojové nástroje. Vzhledem k velkému rozšíření produktů **Microsoft**, konkrétně desktopového operačního systému typu **Microsoft Windows** a serverové edice **Microsoft Windows Server**, jsem zvolila tuto technologii jak jako hostitelský systém pro vývoj a následnou implementaci aplikace tak i jeho vývojový nástroj **Microsoft Visual Studio 2008** a jeho programovací jazyk C# s knihovnou tříd **Microsoft Windows Forms**. Vzhledem k dostupnosti a relativní nenáročnosti na zdroje jsem použila jako databázový systém **Oracle Database10g Express Edition**, v provedení pro 32-bitový operační systém **Microsoft Windows**.

4.1. WINDOWS FORMS

Uživatelské rozhraní je realizováno pomocí knihovny WindowsForms, jež je součástí **Microsoft .NET Framework**. Tato knihovna byla vybrána také z toho důvodu, že umožňuje dynamické rozvržení aplikace a jednoduchou tvorbu ovládacích prvků.

Pro vytvoření uživatelského rozhraní byly použity především kontrolky jako Form, SplitterPanel, TableLayoutPanel, Label, TextBox a Combobox. Všechny nástroje Export, Import, SQL*Loader a Profiler dědí od třídy UserControl, která umožňuje implementaci vlastních kontrol.

4.2. HLAVNÍ TŘÍDA

Základní třídou implementující hlavní formulář a metody pro jeho ovládání je třída OracleTools_Main. Z této třídy se pak inicializují proměnné na výchozí hodnoty, volají se metody zajišťující nastavení pracovní cesty, do které se vytváří všechny pracovní složky a soubory vytvořené během užívání aplikace **OracleVisualTools** kromě. Z této třídy se volá metoda zajišťující otevření formuláře ConnectionManagerForm pro připojení k databázi, dále se odtud zajišťuje otevření záložek pro Export, Import, SQL*Loader a Profiler.

4.3. PŘIPOJENÍ K DATABÁZI

Připojení k databázi je realizováno třídou ConnectionManagerForm vytvářející přihlašovací formulář, žádost o přihlášení se volá metodou Connect. Vytvoří se objekt typu System.Data.OracleClient.OracleConnection, do jehož vlastnosti

ConnectionString se metodou GetDataFromConnectionManagerForm() získají data z formuláře a pomocí třídy ConnectClass se vytvoří požadovaný formát připojovacího řetězce a vloží se do něj hodnoty parametrů UserId, Password, Host, Port a Service_name získané z přihlašovacího formuláře:

Připojovací řetězec:

```
User Id =;Password=;Data
Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=)(PORT=))
CONNECT_DATA=(SERVICE_NAME=))
```

Toto připojení se udržuje po celou dobu běhu aplikace v proměnné Connections.ActiveConnection. Připojení se otevře zavoláním metody Open().

4.4. ČTENÍ HODNOT Z REGISTRU

Součástí analýzy byl poznatek, že pro výběr verze originálního Exportu, Importu a SQL*Loaderu je potřeba znát adresář ORACLE HOME, který získáme z registru z umístění: KEY_LOCAL_MACHINE\SOFTWARE\ORACLE. Následující metoda ukáže, jakým způsobem bylo tohoto cíle dosaženo:

```
{
RegistryKey rk = Registry.LocalMachine;
rk = rk.OpenSubKey("SOFTWARE\\ORACLE");
if (rk == null)
{
throw new UserException("FindOracleHomes()", "", "Directory
KEY_LOCAL_MACHINE\\SOFTWARE\\ORACLE wasn't found in Registry.");
}
string[] folders = rk.GetSubKeyNames();
if (folders.Length == 0)
{
throw new UserException("FindOracleHomes()", "", "No folder in directory
KEY_LOCAL_MACHINE\\SOFTWARE\\ORACLE in Registry.");
}
RegistryKey rkn;
oracle_homes = new List<oracle_home>();
foreach (string folder in folders)
{
rkn = rk.OpenSubKey(folder);
object path = rkn.GetValue("ORACLE_HOME");
object name = rkn.GetValue("ORACLE_HOME_NAME");
if ((path != null) && (name != null))
{
oracle_home oh = new oracle_home();
oh.path = (string)path;
oh.name = (string)name;
oracle_homes.Add(oh);
}
}
}
```

4.5. SOUBORY A ADRESÁŘE

Další důležitou třídou je třída AppData obsahující cestu k výchozímu hlavnímu adresáři aplikace C:\ProgramFiles\OracleVisualTools a implementující metody pro získávání názvů cest potřebných pro běh aplikace odvozováním od cesty hlavního adresáře. Jednou z těchto cest je i pracovní cesta Working directory, kterou může uživatel modifikovat. Tato cesta se ukládá stejně jako jiná nastavení, která mají za úkol přetrvávat i po ukončení aplikace(vyjma uložených connections), do souboru

settings.txt. Tento soubor musí být vždy uložen ve výchozím hlavním adresáři. Tento soubor se vytváří při prvním spuštění aplikace. Formát obsahu souboru je následující:

název parametru | hodnota.

Př.:

Pro získání adresy udump ze souboru settings.txt se použije metoda:

```
string udump = FileClass.GetValueFromSettingsFile(AppData.GetSettingsFile(),  
Constants.udump);
```

Při ukládání hodnot parametrů připojení do databáze vytvoříme nebo přepíšeme tyto hodnoty do souboru connections.txt, který se vytváří vždy přímo do složky WorkingDirecory. Tento soubor obsahuje seznam uložených připojení a má následující strukturu:

>Connection Name | Username | | Host | Port | SID | NetworAlias | Oracle Home
Name | Oracle Home

Velmi důležitým souborem zpracovávaným aplikací OracleVisualTools je soubor tnsnames.ora. Načtení jeho obsahu je důležité ve chvíli, kdy uživatel vybere ORACLE_HOME (všechny ORACLE HOME jsou načteny z registru již při otevírání přihlašovacího formuláře). Podle cesty tohoto parametru jednoduše nalezneme soubor tnsnames.ora. Ten je uložen (pokud existuje) v adresáři ORACLE_HOME\NETWORK\ a obsahuje alias připojení odpovídající připojovacímu řetězci:

```
NA =  
(DESCRIPTION =  
(ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = port))  
(CONNECT_DATA =  
(SERVER = DEDICATED)  
(SERVICE_NAME = service_name)  
)  
)
```

Kde za NA, host, port a service_name jsou uloženy hodnoty těchto parametrů.

Metodou GetTNSDetails() se získávají jednotlivé parametry host, port a service_name ze souboru tnsnames.ora. Metoda se volá poté, co uživatel vybere ORACLE HOME.

4.6. EXPORT, IMPORT, SQL*LOADER

Protože **Export**, **Import** a **SQL*Loader** se liší pouze v názvech parametrů, ale způsob jejich zpracování a ovládání je stejný u všech nástrojů, byla zde snaha implementovat co nejvíce funkčnosti pomocí společného kódu.

Základní třídou pro práci s **Exportem**, **Importem** a **SQL*Loaderem** je třída BaseFullCommand obsahující proměnnou full_command typu Dictionary<string, string>, kde klíčem je název parametru a do jeho hodnoty se ukládají hodnoty zadané uživatelem. Tato třída implementuje metody potřebné pro manipulaci s hodnotami uloženými v proměnné full_command. Od této třídy dědí

třídy ExpFullCommand, ImpFullCommand a SqlldrFullCommand, které při inicializaci zajistí načtení specifických parametrů pro **Export**, **Import** a **SQL*Loader**.

Po kliknutí na tlačítko pro otevření nástroje se vytvoří instance konkrétní třídy-Export, Import nebo SqlLoader. Pro demonstraci vybereme např. třídu Export. Při její inicializaci se vytvoří instance třídy ExpFullCommand a uloží se do proměnné OpendCommandClass.OpendCommand. V této proměnné je udržován odkaz na aktivní proměnnou obsahující seznam parametrů odpovídající otevřenému nástroji. Při otevření dalšího nástroje, např. Importu, se do proměnné OpendCommandClass.OpendCommand uloží instance třídy ImpFullCommand. Seznam všech otevřených nástrojů, tedy instancí tříd ExpFullCommand, ImpFullCommand, SqlldrBaseFullCommand je uložený v proměnné static List<BaseFullCommand> opened_commands třídy OpendCommandClass. Tato třída se stará o správné přepínání mezi proměnnými s uloženými hodnotami parametrů, při uzavření nástroje se stará o vymazání této proměnné ze statické proměnné opened_commands.

4.6.1. Získávání názvů objektů z databáze

Třídy Export, Import a SQLLoader implementují především uživatelské rozhraní a metody pro jeho ovládání. Třídy Export a Import dále obsahují metodu FillPreList, která zajišťuje získávání názvů objektů z databáze. Velmi důležitou třídou pro práci s databází je třída Queries, obsahující všechny SQL příkazy zajišťující funkčnost aplikace.

4.6.2. Spouštění procesu Exportu, Importu a SQL*Loaderu

Pro popis volaných metod vyberu například nástroj **Export**. Při stisknutí tlačítka pro spuštění exportování se spustí metoda void ExportProcessing() třídy Export. Tato metoda zajistí zápis parametrů do parametrického souboru, jelikož je velice pravděpodobné, že by se příkaz nevešel na příkazovou řádku, pomocí metody string SaveCommandToParfile().

Část kódu z této metody zabývající se získáním názvu parametrického souboru:

```
string path_name = AppData.GetExpProcessingParamFile();  
string file_name = path_name + FileClass.GetTimestamp("processing",  
OpenedCommandsClass.OpenedCommand.timestamp, "par");
```

Vytvoří se tedy parametrický soubor s názvem a umístěním:
%WorkingDirctory%\%ConnectionName%\exp\processing\processing_%timestamp%.par.

Části řetězce v [] jsou nahrazeny aktuálními hodnotami. Toto označení se bude vyskytovat v textu i nadále.

Dále se pomocí metody void WriteCommandToBatFile() zavolá:

- 1) metoda string GetCommand(). Tato metoda nejdříve zjistí příkaz, který se má zapsat do souboru,
při OracleHome C:\oracle\app\oracle\product\10.2.0\server:
C:
\oracle\app\oracle\product\10.2.0\server
Exp.exe [username]/[password]@TNS:[TNS] při připojení pomocí TNS nebo
Exp.exe [username]/[password] @//[host]:[port]/[SID] při klasickém připojení.

- 2) Tento řetězec se pro Export uloží do adresáře WorkingDirectory\ConnectionName\exp\bin\exp.bat pomocí metody FileClass.WriteToFile(file_name, command); která se stará o zápis do souboru zadaného textu na zadané místo.
- 3) Následně se spustí metoda void RunBatFile(), která pomocí backgroundworkeru spustí tento batch soubor.

Backgroundworker je třída, která dokáže spustit operaci ve jiném vlákně, než je hlavní aplikace. Pokud by se nespustilo nové vlákno, aplikace by v podstatě nebyla schopna komunikovat s uživatelem, dokud by provádění operace neskončilo. Spouští se metodou RunWorkerAsync() a po ukončení se spustí metoda RunWorkerCompleted().

Po ukončení exportu je batch soubor smazán a uživatel je dotázán, zda chce zadané parametry uložit, což je implementováno pomocí třídy SaveForm.

Pokud ano, parametry se uloží do souboru %WorkingDirectory%\[ConnectionName]\exp\saved\saved_[timestamp].txt.

Timestamp je vytvořena v čase spuštění procesu exportování. Soubor má tvar parametrů v definovaném pořadí, v jakém jsou uloženy v proměnné typu ExpFullCommand. V souboru jsou pak tyto parametry odděleny znakem |. Parametry jsou současně vloženy do ListView pro zobrazování uložených parametrů a pro zobrazování historie Exportu.

4.7. PROFILER

Základem profileru je následující struktura:

```
public struct SessSerId
{
    public string time; //pro uložení času začátku monitorování, v současné verzi se
                        //nevyužívá
    public string sess_id; //session id
    public string serial_id; //serial_id
    public string waits; //priznak, zda sledovat wait event
    public string binds; //priznak, zda vypisovat binds- zatím se nepoužívá se
    public string trace_file; //název trace filu
    public string state; //stav/ nevyužívá se
    public FileStream fs; //pro udržování hodnoty kde jsme skončili pokud je session
                        //monitorovala, následně byla pausnuta a pak zase beží
    public StreamReader sr;
    public BackgroundWorker back_worker; //udržování backworkeru
    public bool broken; //priznak, zda nedošlo k chybě
    public List<Colorize> pattern; //seznamy řetězce a barev pro označování
    public DateTime date_time; //nevyužívá se
    public string timestamp; //pro označení adresáře, do kterého se mají ukládat
                        //soubory vznikající monitorováním
    public string depth; //hloubka, do které se mají dotazy vypisovat
}
```

a seznamy:

```
public static List<SessSerId> SessionsTransfer { set; get; } //session vybraná v
                                                         //prvním formuláři
public static List<SessSerId> WaitingSessions { set; get; } //vybrané sessions z
                                                         //prvního formuláře v
                                                         //menším formuláři
public static List<SessSerId> PreparingSessions { set; get; } //sessions připravující
                                                         //se k monitorování
public static List<SessSerId> TracingSessions { set; get; } //sessions, které se právě
                                                         //monitorují
```

Popis jednotlivých tříd a metod bude uváděn v pořadí jejich volání v průběhu monitorování. Je tak zaručeno, že se pro pois vyberou jen hlavní metody a třídy.

Nejdříve musí uživatel vybrat sessions, které chce monitorovat. To je realizováno pomocí třídy SessionsToTraceForm. Parametry vybraných sessions se uloží do proměnné typu SessSerId a uloží se do listu SessionsTransfer. Tento seznam obsahuje sessions, které byly vybrány uživatelem k monitorování, ale ještě neproběhl kontrola, zda již nebyly vybrány. Po této kontrole se sessions, které byly v pořádku uloží do seznamu WaitingSessions a obsah seznamu SessionsTransfer je smazán. Pokud je již session mezi WaitingSessions nebo TracingSessions, zobrazí se chybové hlášení a vybraná session se odstraní ze seznamu SessionsTransfer. Nyní jsou tedy vybrané sessions v seznamu WaitingSessions a čekají na spuštění monitorování. Spuštění monitorování se provede EventHandlerem private void StartMonitoring_Click(object sender, EventArgs e) zvaným po stisknutí příslušného tlačítka.

Tato metoda opět provede kontrolu, zda se již některé vybrané sessions nemonitorují. Pokud ano, je uživateli zobrazeno chybové hlášení a jsou vyřazeny z WaitingListu. Poté, co proběhne kontrola se provede dotaz do databáze pro zjištění jména trace souboru pomocí metody třídy Queries:

```
public static string GetTraceFileName(string sid, string serial)
{
    string query = " select d.instance_name || '_' || b.trace_file ";
    query += " FROM (SELECT instance_name FROM v$instance) d,";
    query += " (SELECT 'ora_' || spid || (SELECT NVL2(VALUE, '_' || VALUE, NULL) AS VALUE FROM v$parameter";
    query += " WHERE NAME = 'tracefile_identifier') || '.trc' trace_file FROM v$process p, v$session s";
    query += " WHERE (p.addr = s.paddr AND s.sid=" + sid + " AND s.serial#=" + serial + ")) b";
    return query;
}
```

Nezjišťuje se celá cesta, neboť může monitorování probíhat vzdáleně. Získá se pouze hodnota parametru udump ze souboru settings.txt. Jestliže tento parametr není obsahem souboru, získá se hodnota parametru user_dump_dest z pohledu v\$parameter.

Pomocí metody private void GetNamesOfTraceFiles() se k jednotlivým sessions přiřadí jména jejich trace souborů:

```
string sid.trace_file = Path.Combine(user_dump_dest_label.Text, RunQueryToGetTraceFileName(query));
```

Kde metoda

```
string query = Queries.GetTraceFileName(sid.sess_id, sid.serial_id);
public static string GetTraceFileName(string sid, string serial)
{
    string query = " select d.instance_name || '_' || b.trace_file ";
    query += " FROM (SELECT instance_name FROM v$instance) d,";
    query += " (SELECT 'ora_' || spid || (SELECT NVL2(VALUE, '_' || VALUE, NULL) AS VALUE FROM v$parameter";
    query += " WHERE NAME = 'tracefile_identifier') || '.trc' trace_file FROM v$process p, v$session s";
    query += " WHERE (p.addr = s.paddr AND s.sid=" + sid + " AND s.serial#=" + serial + ")) b";
    return query;
}
```

Získá pouze jméno trace souboru a jeho cesta je doplněna hodnotou z labelu, který může uživatel editovat a nastavit aby mohl sledovat monitorování na jiném stroji

než je nainstalována databáze. Když už známe názvy trace souborů, může se spustit SQL příkaz pro spuštění dbms_monitoru pomocí metody RunMonitoringQuery(string query).

Třída Tracing implementuje metody pro práci s trace soubory. Zavoláním metody Tracing.StartReadingFromTraceFiles(); se začne číst z trace souborů, které jsou již vytvořeny a StreamReader, jež je uložený jako jedna z vlastností v session, se nastaví na jejich konec. Pro uložení kde právě v trace souboru skončilo předchozí monitorování a nastavení se na nové místo slouží StreamReader. Třída ProfilerBackgroundWorkers pak provádí poslední nastavení před spuštěním monitorování, jako je například uložení řetězců a k nim nastavených barev pro zvýrazňování textu a inicializování backgroundWorkerů.

Spuštěním metody

```
static void back_worker_DoWork(object sender,
System.ComponentModel.DoWorkEventArgs e) začne pokus o čtení dat z trace
souboru, jehož cesta je uložena v proměnné session. Tato session byla předána do
metody jako argument. Vlákno se tedy pokouší otevřít soubor uložený v proměnné
session pomocí metody Tracing.OpenTraceFiles(session). Soubor se však vytvoří až
poté, co je do databáze pro danou session odeslán nějaký SQL příkaz. Do té doby je
v metodě nastavena smyčka testující, zdali byl již soubor vytvořen nebo zda má na
jeho vytvoření ještě čekat.:
```

```
while (!File.Exists(path))
{
    Thread.Sleep(1000);
}
```

Poté, co se je vytvořen trace soubor a je otevřen příslušným vláknem začíná while smyčka čtení ze souboru.

Formát trace souboru:

```
Dump file c:\oracle\app\oracle\admin\xe\udump\xe_ora_2704.trc
Sun Aug 02 15:20:16 2009
ORACLE V10.2.0.1.0 - Production vsnsta=0
vsnsql=14 vsnxtr=3
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
Windows XP Version V5.1 Service Pack 2
CPU : 2 - type 586
Process Affinity : 0x00000000
Memory (Avail/Total): Ph:150M/1022M, Ph+PgF:1119M/2456M, VA:1606M/2047M
Instance name: xe

Redo thread mounted by this instance: 1

Oracle process number: 20

Windows thread id: 2704, image: ORACLE.EXE (SHAD)

*** 2009-08-02 15:20:16.481
*** SERVICE NAME:(SYS$USERS) 2009-08-02 15:20:16.309
*** CLIENT ID:(eva:127.0.0.1) 2009-08-02 15:20:16.309
*** SESSION ID:(31.257) 2009-08-02 15:20:16.309
=====
PARSING IN CURSOR #2 len=37 dep=1 uid=0 oct=3 lid=0 tim=107681468090 hv=1398610540
ad='20ae9760'
select text from view$ where rowid=:1
END OF STMT
PARSE #2:c=0,e=116477,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681468075
EXEC #2:c=0,e=110245,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681798557
FETCH #2:c=0,e=89,p=0,cr=2,cu=0,mis=0,r=1,dep=1,og=4,tim=107681803410
```



```
STAT #2 id=1 cnt=1 pid=0 pos=1 obj=63 op='TABLE ACCESS BY USER ROWID VIEW$ (cr=1 pr=0
pw=0 time=71 us)'
=====
```

Informace týkající se jednoho SQL příkazu jsou ohraničeny řetězcem "===== ". Na začátku monitorování se do trace souboru zapíše obecné informace, které nepotřebujeme zobrazovat. Dále jsou jednotlivé řádky souboru vyskytující se mezi těmito dvěma řetězci ukládány do proměnné `List<string> lines`. Jakmile jsou načteny všechny řádky jednoho příkazu, je spuštěna metoda `ProcessTrace.ProcessOneTrace(lines, part, end, session)` zpracovávající řádky se statistikami jednoho SQL příkazu. Struktura, která se využívá pro uložení všech hodnot SQL příkazu:

```
struct OneTrace
{
    public List<OnePar> parsing;
    public List<string> statement;
    public List<List<OnePar>> parse;
    public List<List<OnePar>> exec;
    public List<List<OnePar>> fetch;
    public List<List<OnePar>> call;
    public List<string> call_names;
    public List<List<OnePar>> stat;
    public List<List<OnePar>> wait;
    public List<string> binds;
}

struct OnePar
{
    public string name;
    public string sc; //shortcut
    public string value;
}

class BaseTracePars
{
    protected string What{set;get;}
    protected List<string> names;
    protected List<string> sc;
    protected List<OnePar> tr_pars;

    public BaseTracePars()
    {
        What = "none";
    }
}

class ParsingPars : BaseTracePars
{
    public ParsingPars()
    {
        What = "PARSING";
        names = new List<string>();
        sc = new List<string>();
        names.AddRange(new string[] { "cursor", "len", "dep", "uid", "oct", "lid",
                                         "tim", "hv", "ad" });
        sc.AddRange(new string[] { "cursor", "len", "dep", "uid", "oct", "lid",
                                     "tim", "hv", "ad" });
        MergeTwoLists(names, sc);
    }
}
```

Třída `ProcessTrace` se stará o vyfiltrování hodnot z textu a jejich přiřazením ke správným parametrům do proměnné typu `OneTrace`.

Třída `ShowTraceParams` se pak stará o zobrazení příslušných hodnot. Jelikož se však pohybujeme ve vláknech a potřebujeme zapisovat do `ListView monitoring_lv` společného všem vláknům, je tento problém vyřešen pomocí delegátů a metody `Invoke()`.

Metoda Invoke() zavolá specifikovaného delegáta obsluhujícího monitong_lv spolu s metodou pro přidání položek. Jako parametr předá položky, které potřebuje přidat do monitong_lv.

Pro hlídání počtu položek v monitoring_lv je určena proměnná int ProfilerCurrent.TracesCount, která udržuje hodnotu aktuálního počtu položek v monitoring_lv.

O zapsání SQL příkazu a jeho statistik do souboru, se kterým bude následně pracovat aplikace OracleVisualTools při zobrazování historie se stará metoda public static void SaveTraceParamsToFile(string filepath, ListView.ColumnHeaderCollection headers, List<ListViewItem>items) třídy FileClass.

Hodnoty jednotlivých parametrů jsou uloženy ve formátu:

```
timestamp;TIMESTAMP;
session_id;SID;
serial_id;SERIAL;
statement;STATEMENT;
call;CALL;
#;CURSOR;
c;c;
e;e;
p;p;
cr;cr;
cu;cu;
mis;mis;
r;r;
dep;dep;
og;og;
tim;tim;
operations;OPERATIONS;
wait_events;WAIT EVENTS;
***
>7.12.2009      17:32:05;27;18;select;Parse;;1;281250;722043;0;16;0;1;0;0;1;7.12.2009
17:32:05;;'SQL*Net;
>;; distinct granted_role;Execute;;1;0;454;0;0;0;0;0;1;7.12.2009 17:32:05;;
```

Kdy řádky uložené na začátku souboru po znak *** obsahují po řadě název ColumnHeader ch.Name a název sloupce, který vidí uživatel.

Řádky za symbolem *** pak reprezentují vždy jednu položku monitoring_lv. Jelikož se však v průběhu monitorování vytvoří více souborů ve složce Output\profiler\sessions\trace\[timestamp]\, jsou názvy sloupců uloženy pouze v prvním z těchto souborů Output\profiler\sessions\trace\[timestamp]\0.txt. Názvy souborů jsou tvořeny pořadovým číslem nastaveným podle toho, kolikrát monitoring_lv bylo vyplněno.

4.7.1. TKProf

Příprava souborů a vlastní spuštění souboru začíná pomocí metody

```
private void tkprof_button_Click(object sender, EventArgs e),
```

která je součástí třídy Profiler.

Tato metoda nejdříve zajistí zřetězení souborů do jednoho pomocí metody string file = TKProf.GetOneTraceFile(to) kde to je proměnná typu List<SessSerId>. Jde tedy o seznam s uloženými jednotlivými session. Tyto session obsahují název a umístění trace souboru. Výsledný soubor bude mít název:

```
%Working directory%\[Connection_name]\profiler\sessions\tkprof\concat_[timestamp].trc.
```

Dále se vytvoří string newfile, do kterého se uloží výstup nástroje tkprof.

```
%Working directory%\[Connection_name]\profiler\sessions\tkprof\tkprof_[timestamp].txt.
```

Následně se vytvoří batch soubor názvu:

%Working directory%\[Connection_name]\profiler\sessions\tkprof\mytkprof.bat
do kterého se uloží řetězec pro spuštění nástroje **TKProf**.

Soubor tedy bude mít pro ORACLEHOME
C:\oraclexe\app\oracle\product\10.2.0\server

Následující formát:

C:
\oraclexe\app\oracle\product\10.2.0\server\bin
Tkprof.exe %file% %newfile%

Metoda TKProf.ProcessWordPad(newfile) spustí vykonávání nástroje **TKProf**
a výsledek zobrazí ve WordPadu.

5. UŽIVATELSKÁ DOKUMENTACE

5.1. ZÁKLADNÍ FUNKCE PROGRAMU

Aplikace **OracleVisualTools** (dále **OVT**) nabízí uživatelské rozhraní pro originální nástroje **Oracle Database 9i, 10g a 11g** a to **Export, Import a SQL*Loader²**. Dále nabízí možnost monitorování SQL příkazů a jejich statistik **Oracle Database 10g** pomocí nástroje Profiler. Vše musí být instalováno v prostředí **Microsoft Windows®**, odladěno na 32-bit verzi

5.2. MINIMÁLNÍ KONFIGURACE

OS MS Windows XP SP2 nebo SP3

Microsoft .Net Framework 3.5

Microsoft Windows Installer 2.0 nebo vyšší

Alespoň jedna z verzí **Oracle Database 9i, 10g, 11g** nebo alespoň jedna z verzí **Oracle Database Client 9i, 10g, 11g** v custom nebo administrátorské instalaci

Pro Profiler se vyžaduje **Oracle Database 10g** běžící v prostředí **Microsoft Windows®**

5.3. NUTNÁ OPRÁVNĚNÍ

Databázový systém **Oracle** :

Databázové utility **Export , Import** - právo EXP_FULL_DATABASE, IMP_FULL_DATABASE pro databázové objekty jichž uživatel není vlastníkem

Pro Profiler se vyžaduje uživatelský účet s rolí dba nebo alespoň s právy EXECUTE na balíčkem **dbms_monitor**

Bližší informace k právům jsou uvedeny dále v textu

Práva v hostitelském operačním systému **Microsoft Windows®**

Právo uživatele na instalaci SW

Právo uživatele na hostiteli **Oracle Database** na čtení souborů *.trc z adresáře \udump kam ukládá **dbms_monitor** výstup (informace o adresáři na hostitelském serveru je podána uživateli aplikací **OracleVisualTools**). V případě spouštění z klientské stanice na síti je nutné mít adresář namapovaný na síťovou jednotku také s právem čtení souborů *.trc

5.4. INSTALACE

Instalace probíhá klasicky pomocí instalačního balíčku Setup.msi. Po dvojkliku na tento soubor se spustí instalační služba **Microsoft Windows Installer**, pomocí

² Popis jejich parametrů se nachází v příloze DOPLNIT UMÍSTĚNÍ

kterého se spustí průvodce instalací. V tomto průvodci se lze posouvat pomocí tlačítka Next. Uživateli je zde nabídnuto zvolit jiné než výchozí umístění pro spustitelný soubor **OracleVisualTools.exe**. Při instalačním procesu se také kontroluje zda je nainstalovaný balíček **Microsoft .Net Framework 3.5** SP1. Pokud tomu tak není, provede se instalace tohoto balíčku.

V případě existence starší verze Oracle VisualTools instalační balíček tuto verzi indikuje. Do té doby, než uživatel starší verze neodinstaluje, instalace nové verze není možná. Po úspěšném nainstalování aplikace se ve zvoleném adresáři vytvoří spustitelný soubor **OracleVisualTools.exe**, dále se vytvoří zástupce na ploše a v nabídce Start ve složce OracleVisualTools.

5.5. SPUŠTĚNÍ APLIKACE

Aplikaci lze spustit kliknutím na soubor **OracleVisualTools.exe** v umístění zvoleném při instalaci nebo kliknutím na příslušného zástupce.

Po spuštění tohoto souboru se otevře hlavní okno aplikace. Toto okno je rozděleno na tři části- menu v horní části, strom s připojeními v levé části a hlavní část okna zabírá prostor pro práci s **Exportem, Importem, SQL*Loaderem a Profilerem**. Menu obsahuje záložky File, Tools a Options.

Pomocí tlačítka New connection v záložce File se zobrazí formulář Connection Manager Form pomocí něhož lze vytvořit a uložit nové připojení a připojit se k databázi. Kliknutí na tlačítko Disconnect způsobí odpojení od databáze, pokud nějaké spojení existovalo a pomocí tlačítka Exit se zavře aplikace. Pokud existovalo připojení k databázi, před uzavřením formuláře se toto spojení uzavře.

Záložka Tools nabízí spuštění odpovídajících nástrojů stejně jako tlačítka exp, imp, sql*ldr a profiler pod lištou menu.

Záložka Options nabízí tlačítko Working Directory pro změnu pracovního adresáře.

řádky nebudou v korektním stavu, pak aplikace OracleVisualTools tyto řádky při následujícím spuštění smaže. Popis souboru connections.txt:

V souboru jsou uložena všechna připojení, která uživatel aplikace OracleVisualTools nadefinoval a uložil, ve formátu:

```
>Connection Name|Username||Hostname|Port|SID|Network Alias|Oracle Home Name|  
Oracle Home Path
```

Jednotlivé parametry jsou popsány níže.

Druhou částí je Oracle Connection. Zde se nastavují parametry připojení:

- ☐ Username- uživatelské jméno
- ☐ Password- heslo pro dané uživatelské jméno
- ☐ Hostname- název nebo přímo IP adresa serveru, kde je DBS Oracle spuštěn
- ☐ Port- číslo portu, přes který DBS Oracle komunikuje
- ☐ SID- service name- název služby, která jednoznačně identifikuje instanci a databázi

Další parametry:

- ☐ Connection Name- název připojení
- ☐ Oracle Home Name- identifikátor Oracle Home, specifikuje skupinu programů a služeb k příslušnému Oracle Home
- ☐ Oracle Home- adresář, ve kterém jsou nainstalované produkty Oracle
- ☐ Connection Type- typy připojení
- ☐ Network Alias- alias připojení nakonfigurovaného v konfiguračním souboru tnsnames.ora dostupného jak na serveru tak na klientovi.

Pro úspěšné připojení k databázi je nutné nastavit parametry připojení a dále je nutné z nabídky Oracle Home Name zvolit verzi databáze/klienta, neboť na jednom počítači je možné mít nainstalováno více těchto verzí. Výběr Oracle Home Name je důležitý jak pro připojení k databázi pomocí TNS popsaném níže, tak ke zvolení nástrojů správné verze databáze/klienta při práci s exportem, importem, sql*loaderem a profilerem. Po zvolení Oracle Home Name se v parametru Oracle Home zobrazí příslušná cesta.

Typy připojení jsou možné dva a to Basic a TNS. Výchozím typem je typ Basic. Při zvolení tohoto parametru je nutné vyplnit parametry Hostname, Port a SID.

Při připojení typem TNS seznam všech adres připojení spolu s jejich aliasy obsahuje konfigurační soubor tnsnames.ora buď na straně klienta nebo na serveru, záleží na tom odkud se připojujeme. Díky tomuto souboru stačí pouze vybrat alias připojení a není nutné vyplňovat hodnoty parametrů Hostname, Port a Sid.

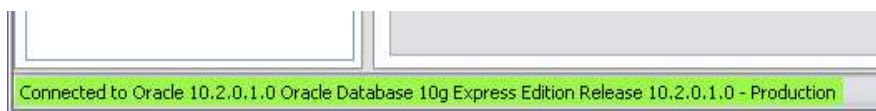
Při zvolení připojení typu TNS je však nutné mít předem vybranou hodnotu parametru Oracle Home Name. Pokud je tato podmínka splněna, zobrazí se v nabídce parametru Network alias aliasy pro již nadefinovaná připojení z konfiguračního souboru tnsnames.ora vybraného podle cesty parametru Oracle Home.

Tento soubor nemusí existovat a pokud by jej chtěl uživatel využívat, musí si jej vytvořit sám vložením následujícího řetězce a doplněním hodnot na místa NA, protocol, host, port, service_name

```
NA =
(DESCRIPTION =
  (ADDRESS =
    (PROTOCOL = protocol)
    (HOST = host)
    (PORT = port)
  )
  (CONNECT_DATA =
    (SERVICE_NAME = service_name)
  )
)
```

Nový soubor nazvaný tnsnames.ora je nutné umístit do adresáře %ORACLE_HOME%\NETWORK\ADMIN\ . Toto je defaultní adresář, ve kterém hledá soubor tnsnames.ora aplikace **OracleVisualTools**. Je také možné, že cesta k tnsnames.ora byla změněna. V takovém případě by byla uložena v proměnné tns_admin přítomné buď v registrech nebo v proměnné prostředí. Aplikace **OracleVisualTools** soubor tnsnames.ora používá, nikoliv vytváří. Konkrétní připojení vybereme zvolením příslušného aliasu, jehož hodnoty je možné vidět u parametrů Hostname, Port a SID. Dalším parametrem je Connection Name. Tento parametr je důležitý pro aplikaci Oracle Visual Tools pro identifikaci připojení. Jeho nastavení je vyžadováno a musí být unikátní.

Hodnoty parametrů můžeme kdykoliv uložit pomocí tlačítka Save, vždy se vyžaduje alespoň zadání názvu připojení. Pokud je kontrola vstupních parametrů v pořádku, uloží se nové připojení do souboru connections.txt a dále se ihned toto připojení zobrazí ve stromě připojení. Tlačítko Clear uvede formulář do výchozího stavu a tlačítko Test testuje, zda je připojení korektní. Výsledek testu se zobrazí v řádku se statusem nad tlačítky. Proces připojení k databázi se realizuje pomocí tlačítka Connect. Jestliže bylo připojení úspěšné, formulář se uzavře a do stavového řádku ve spodní části hlavního okna se zobrazí název databáze a obrázek u aktivního připojení se označí zeleně. V opačném případě se ve statusu zobrazí chybové hlášení. Pokud právě probíhá pokus o připojení, lze jej přerušit stisknutím tlačítka Cancel. To však vyvolává i zavření celého formuláře.



OBRÁZEK 5: STAVOVÝ ŘÁDEK VE SPODNÍ ČÁSTI HLAVNÍHO FORMULÁŘE ZOBRAZUJÍCÍ STAV PŘIPOJENÍ A EDICI A VERZI DATABÁZE

Upozornění: Při používání nástrojů z verze Oracle Database 9i je nutné zadávat přihlašovací údaje pouze formou TNS. Důvodem je jiný způsob zpracování přihlašovacího řetězce používaný u Oracle nástrojů verze 9i než ve verzích Oracle Database 10g nebo 11g.

Při pokusu o zobrazení Connection ManagerForm se mohou zobrazit tyto chybová hlášení:

- 1) Directory KEY_LOCAL_MACHINE\SOFTWARE\ORACLE wasn't found in Registry.- zřejmě nemáte nainstalovaný Oracle.
- 2) No folder in directory KEY_LOCAL_MACHINE\SOFTWARE\ORACLE in Registry- v registrech v uvedené cestě nebyla nalezena žádná složka- zřejmě nemáte nainstalovaný Oracle.
- 3) No ORACLE HOME wasn't found in any subfolder of folder KEY_LOCAL_MACHINE\SOFTWARE\ORACLE\%Folder%\ in Registry.- v uvedené cestě nebyl nalezen klíč ORACLE HOME a ORACLE HOME NAME.

5.7. ODPOJENÍ OD DATABÁZE

Pro opojení od databáze slouží tlačítko Disconnect v záložce menu File. Odpojit se od databáze lze také pravým kliknutím myši na příslušné připojení a výběrem položky Disconnect. Násedkem odpojení se vypíše do stavového řádku hláška "Disconnected".

5.8. ZMĚNA PRACOVNÍHO ADRESÁŘE

Výběrem položky Working Directory se zobrazí formulář pro změnu pracovního adresáře (nazvaný Output). Změnu tohoto adresáře je možné provést po kliknutí na tlačítko Browse. Tlačítkem Default se nastaví výchozí adresář. Tlačítka OK a Cancel mají klasický význam.

Při změně pracovního adresáře dochází k přesunu všech složek, podsložek a souborů vytvořených při práci s aplikací, jako jsou například pomocné soubory při práci s profilerem nebo výchozí adresáře pro ukládání souborů vytvořených **Exportem, Importem a SQL*Loaderem**. Nepřenáší se pouze soubor settings.txt. Tento soubor obsahuje hodnotu parametru Working Directory a je vždy umístěn ve výchozím adresáři. Pokud dojde ke smazání nebo přesunutí tohoto souboru, po zadání pracovního adresáře a potvrzením tlačítkem OK aplikace vytvoří pracovní adresář a soubor settings.txt ve výchozím adresáři a vytvoří novou složku pro ukládání souborů vytvořených uživatelem při práci s aplikací.

5.9. OVLÁDÁNÍ NÁSTROJŮ

V následujících řádcích lze nalézt popis používání nástroje **Export**. V kapitole **Import** a **SQL*Loader** jsou popsány hlavní rozdíly oproti nástroji **Export**, neboť jsou si tyto nástroje velmi podobné a ovládání je analogické, není těmto nástrojům dále věnována větší pozornost. Aplikace **OracleVisualTools** využívá originální **Export**, originální **Importu** a originální **SQL*Loader**. Jejich verze se nastavuje podle proměnné Oracle Home zadávané při připojení. Při práci s jednotlivými nástroji se soubory vytvářené těmito nástroji ukládají do pracovního adresáře:

%Working direcorey%\[název připojení]\[název nástroje].

```

User\[Connection Name]\exp\dump
      \log
      \parfile
      \processing
      \saved
      \imp\log
      \parfile
      \processing
      \saved
      \sqlldr\bad
            \controlfile
            \data
            \discard
            \indexfile
            \log
            \parfile
            \processing
            \saved
      \profiler\sessions\trace\[timestamp]
            \tkprof

```

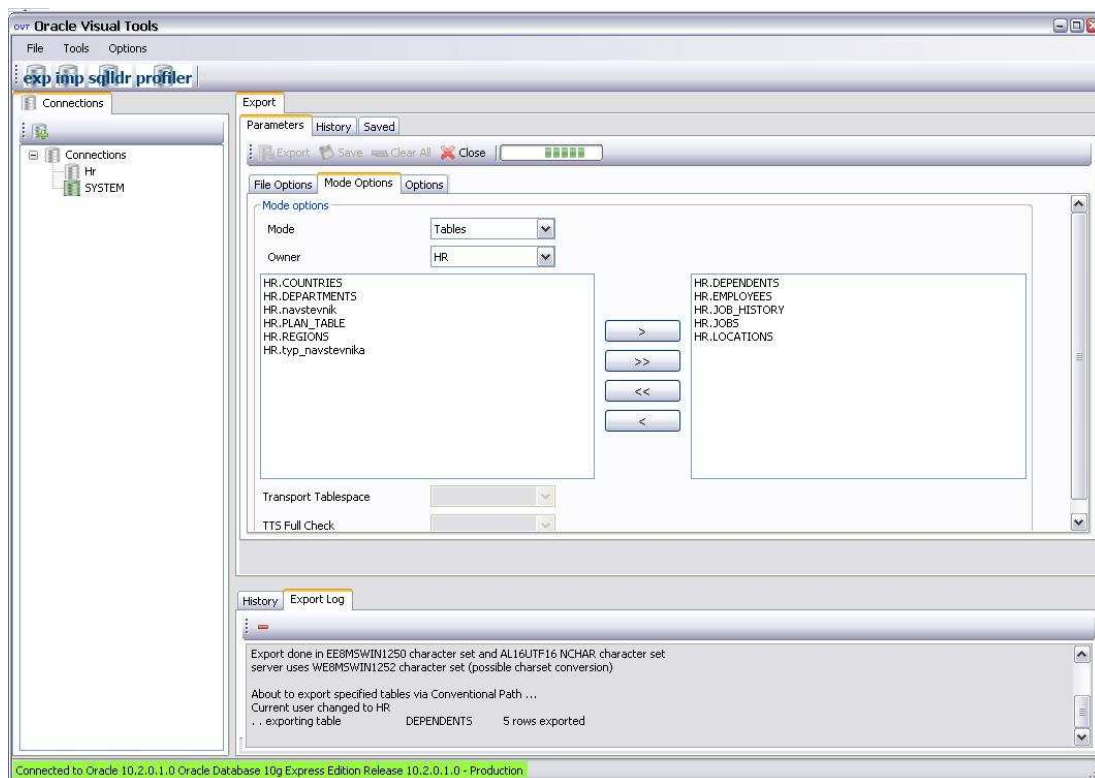
OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA

Umístění těchto souborů je patrné z adresářové struktury viz.

OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA.

Po otevření např. **Exportu** lze rovněž otevřít **Import**, **SQL*Loader** a **Profiler**, nelze však otevřít více záložek stejného nástroje. Mezi jednotlivými záložkami se lze libovolně přepínat.

Popisy parametrů jednotlivých nástrojů jsou uvedeny v příloze.



OBRÁZEK 7:EXPORT

5.9.1. Export

Tento nástroj nabízí možnost nastavení parametrů Exportu, uložení těchto parametrů, jejich opětovné obnovení a dále pak samotný proces exportování dat z databáze, ke které je uživatel momentálně připojen. Verze originálního nástroje pro Export, které využívá aplikace OracleVisualTools je určena cestou v Oracle Home zadané uživatelem při nastavování připojení k databázi.

Omezení:

- ❑ Práva CREATE SESSION pro export objektů vlastního schéma
- ❑ Práva EXP_FULL_DATABASE pro export objektů vlastněných jiným uživatelem

Následující schéma nemohou být exportována:

- ❑ ORDSYS
- ❑ MDSYS
- ❑ CTXSYS
- ❑ ORDPLUGINS
- ❑ LBACSYS

Umístění souborů spojených s Exportem

Umístění souborů spojených s Exportem je patrné z adresářové struktury viz OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA

Otevření

Tento nástroj lze otevřít buď vybráním položky v záložce *Tools* nebo kliknutím na tlačítko pod lištou menu. Jestliže v daný moment neexistuje připojení k databázi, zobrazí se formulář pro správu připojení. Po úspěšném připojení se zobrazí záložka Exportu stejně jako v případě, kdy by v době kliknutí na vybraný nástroj připojení existovalo. V případě stisknutí tlačítka Cancel ve formuláři pro správu připojení se tento formulář zavře a záložka Exportu se neobjeví.

Použití:

Nástroj **Export** umožňuje prohlížení aktuálně zadávaných parametrů, historie a uložených parametrů pomocí záložek Parameters, History a Saved.

Parameters

Záložka Parameters je hlavní záložka pro práci s **Exportem**. V horní části této záložky se nachází lišta s tlačítky

- ❑ Export pro spuštění procesu exportování dat z databáze
- ❑ Save pro uložení hodnot parametrů, které má uživatel aktuálně nastavené
- ❑ Clear pro smazání aktuálně nastavených hodnot
- ❑ Close pro uzavření celé záložky Exportu. Toto tlačítko však nezastaví právě probíhající proces Exportu.

Největší část této záložky však pokrývá množství parametrů **Exportu**, které lze nastavit standardními ovládacími prvky. Tyto parametry jsou rozděleny podle souvislosti do dalších tří záložek: File options, Mode options a Options.

File Options obsahuje parametry pro práci se soubory, lze zde také nastavit možnost vystupovat jako SYSDBA. SYSDBA má speciální funkce. Jeho chování není stejné jako všeobecných uživatelů. Měla by se použít v případě parametru transportable tablespaces.

Export nevyžaduje zadat žádné vstupní názvy souborů, uživatel však může zadat parametrický soubor (Parameter File) s již uloženými vstupními parametry pro **Export**. Pokud následně dojde k zadání stejných nebo jiných hodnot stejných parametrů jako jsou v parametrickém souboru, použijí se hodnoty nově zadané v nástroji **Export**.

Výstupním souborem je exportní soubor s příponou.dmp. Tento soubor je vytvořen originálním **Exportem** a je možné jej opět importovat do databáze. Dále je vytvořen soubor s protokolem o průběhu exportování (Log File).

Pokud uživatel nevyplní název exportního souboru a souboru s protokolem, jsou tyto hodnoty nastaveny na tvar dump_[časové razítko].dmp pro exportní soubor a log_[časové razítko].log pro soubor s protokolem a uloženy v adresářové struktuře viz Obrázek 6: Adresářová struktura.

Záložka Mode options umožňuje uživateli vybrat objekty, které chce z databáze exportovat. Lze zde nastavit mód **Exportu** a to jeden z následujících: Full, Owners, Tables, Tablespaces.

Co umožňují exportovat následující módy:

- ☐ Full -všechny objekty z databáze
- ☐ Owners- objekty vybraných vlastníků
- ☐ Tables- tabulky
- ☐ Tablespaces- tabulkové prostory

Pro zjednodušení práce je po vybrání jednoho z módů Owners, Tables nebo Tablespaces nabídnut seznam uživatelských jmen vlastníků, seznam tabulek nebo tabulkových prostorů v levém okénku, ze kterého uživatel zvolí položky, které chce exportovat, pomocí šipek >, >>, <, << do pravého okénka.

Záložka Options obsahuje seznam ostatních parametrů. Nastavení hodnot těchto parametrů je standardní.

Uložení hodnot parametrů

Ve většině případů uživatel při exporu zadává stejné nebo velmi podobné hodnoty parametrů. Z tohoto důvodu je zde nabídnuta možnost si tyto parametry uložit pomocí stisknutí tlačítka Save. Hodnoty těchto parametrů je možné si prohlédnout, smazat, popřípadě obnovit uložené parametry v záložce Saved popsané níže.

History

Záložka History nabízí pohled na datum, čas a hodnoty parametrů úspěšných procesů exportu realizované od spuštění aplikace. Všechny tyto hodnoty lze smazat kliknutím na tlačítko ClearAll.

Saved

Záložka Saved jako bylo napsáno výše poskytuje prohlédnutí, smazání, popřípadě obnovení uložených parametrů. Každému uloženému záznamu odpovídá jedna řádka. Smazat lze viz záznamů najednou jejich označením a stisknutím tlačítka Clear. Stisknutím tlačítka ClearAll dojde ke smazání všech záznamů a stisknutím tlačítka Restore k jejich obnovení vložením hodnot k příslušným parametrům v záložce Saved.

Spuštění procesu Export

Samotný proces exportování zadaných parametrů se spustí po stisknutí tlačítka Export. Aplikace **OracleVisualTools** vygeneruje spouštěcí dávku obsahující příkaz exp.exe spouštěný z adresáře Oracle Home\bin\ se zadanými parametry exportu a přihlašovacím řetězcem podle zvoleného připojení. Dávka je předána operačnímu systému k provedení tak, že běží v samostatném vlákně.

Po stisknutí tlačítka se uživateli zobrazí formulář Edit, kde má uživatel poslední možnost změnit parametry. Edit je zde nabídnuto z toho důvodu, pokud by chtěl například využít pattern matching, příklad:

```
TABLES=(scott.%P%,blake.%,scott.%S%)
```

Nebo v případě, že chce uživatel exportovat pouze určité partitions:

```
TABLES=(emp:m)
```

Ne vždy se však podaří uskutečnit Export bez jakýchkoliv problémů. O celém průběhu exportování a o těchto problémech je uživatel průběžně informován v záložce ExportLog viz. OBRÁZEK 7:EXPORT.

Výsledek procesu Export

Bezprostředně po skončení procesu se zobrazí uživateli formulář s dotazem, zda chce uložit hodnoty parametrů, které zadal při tomto Exportu.

Po skončení procesu se do záložky History vypíše datum a čas provedeno Exportu spolu s hlášením, zda proces proběhl úspěšně nebo neúspěšně. Dále je vytvořen exportní soubor a soubor s protokolem, které zadal uživatel. V případě že tyto soubory zadány nebyly, lze tyto soubory nalézt v adresářové struktuře viz OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA.

5.9.2. Import

Tato kapitola popisuje rozdíly oproti **Exportu**. Pokud zde tedy nenaleznete kapitolu nebo podkapitolu kterou hledáte, podívejte se prosím do příslušné kapitoly Exportu.

Omezení:

- ☐ Je nezbytné mít právo CREATE_SESSION
- ☐ Pokud uživatel A vytvoří exportní soubor, uživatel B jej může importovat za podmínky IMP_FULL_DATABASE role

- ❑ Jestliže exportní soubor vytvořil uživatel s rolí DBA, pak tento soubor může importovat pouze uživatel s rolí IMP_FULL_DATABASE
- ❑ Pro mód importu Full musí mít uživatel právo IMP_FULL_DATABASE
- ❑ Pro import práv spolu s objekty musí uživatel spouštějící Import buď vlastnit tento objekt nebo musí mít právo WITH_GRANT_OPTION
- ❑ Importovat lze pouze soubory vytvořené nižší nebo stejnou verzí originálního Exportu jako je originální Import. Verze originálního Exportu a originálního Importu se nastavuje podle proměnné Oracle Home zadávané při připojení.

Použití

Nástroj Import umožňuje prohlížení aktuálně zadávaných parametrů, historie a uložených parametrů pomocí záložek *Parameters*, *History* a *Saved*.

Parameters

Záložka File Options obsahuje parametry pro práci se soubory, lze zde také nastavit možnost vystupovat jako SYSDBA.

Import vyžaduje vstupní parametr se souborem Import file s příponou .dmp vytvořený originálním **Exportem**. Tento název lze zadat ve formuláři nebo může být součástí parametrického souboru (Parameter File) s již uloženými vstupními parametry. Zde stejně jako u Exportu v případě, že dojde k zadání stejných nebo jiných hodnot stejných parametrů jako jsou v parametrickém souboru, použijí se hodnoty nově zadané v nástroji **Import**.

Výstupním souborem je soubor s protokolem o průběhu Importu (Log File). Volitelně mohou být zadány soubory Data Files (zatím je umožněno zadání pouze jednoho souboru, zadání více souborů se plánuje jako rozšíření) a Index File. Popisy těchto souborů jsou uvedeny v příloze.

Pokud nebyl vyplněn název importního souboru, bude na to uživatel upozorněn při stisknutí tlačítka Import pro spuštění procesu Importu. Pokud nebyl zadán název souboru s protokolem, je tento název nastaven na log_[časové razítko].log a je uložen v adresářové struktuře viz. OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA.

Záložka Mode options umožňuje uživateli vybrat objekty ze souboru Import File, které chce importovat do databáze. Lze zde nastavit mód Importu, a to jeden z následujících: Full, FromUser, Tables, Tablespaces.

Co umožňují importovat následující módy:

- ❑ Full - celý exportní soubor
- ❑ FromUser- všechny objekty(tabulky, data, práva, indexy) jednoho nebo více vlastníků jiným vlastníkům
- ❑ Tables- tabulky
- ❑ Tablespaces- tabulkové prostory

Uživatel sice může importovat pouze ty objekty, které jsou uloženy v exportním souboru, při výběru módu jsou mu však nabídnuty všechny objekty připojené databáze, ale importují se pouze ty, které jsou v exportním souboru a současně ve výběru uživatele.

Import na rozdíl od **Exportu** nabízí okének pro výběr objektů více- pro módy FromUser, Tables a Tablespace. Objekty z připojené databáze jsou vždy načteny do levého okénka a uživatel vytváří výběr přesouváním objektů do pravého okénka pomocí šipek.

Jestliže je zvolen mód FromUser, zobrazí se v levém okénku From user seznam uživatelských jmen vlastníků. Vybrání vlastníků, jejichž objekty chce uživatel importovat provede pomocí tlačítek >, >>, <, <<.

V okénkách To user stejným způsobem uživatel vybere uživatelská jména cílových vlastníků.

Jestliže chce uživatel importovat pouze konkrétní tabulky, lze toto vykonat dvěma způsoby. Jedním způsobem je vybrání módu Tables a poté z okénka Tables vybrat konkrétní tabulky. Druhý způsob nabízí mód FromUser, kde uživatel vybere z okénka From user jednoho vlastníka, jehož tabulky pomocí tlačítka Insert vloží do okénka Tables a konkrétní tabulky pro importování lze z tohoto okénka vybrat výše popsaným způsobem.

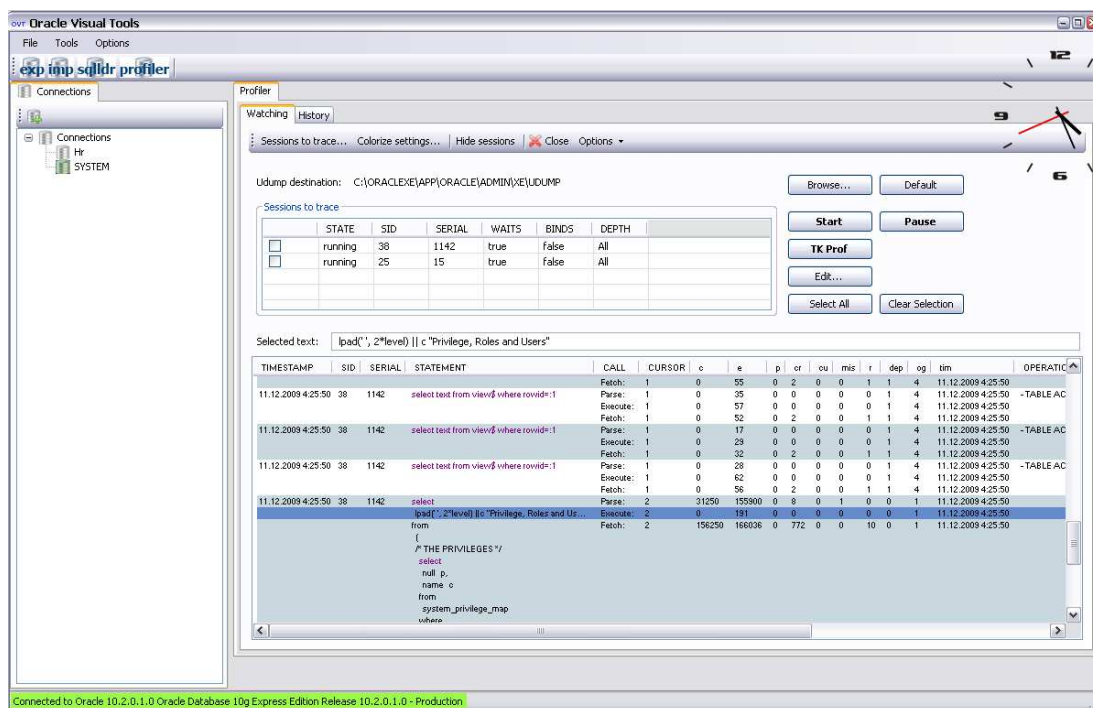
5.9.3. SQL*Loader

SQL*Loader vkládá data z externích souborů do tabulek databáze Oracle.

Tato kapitola popisuje rozdíly oproti **Exportu**. Pokud zde tedy naneleznete kapitolu nebo podkapitolu kterou hledáte, podívejte se prosím do příslušné kapitoly **Exportu**.

Omezení:

- ❑ Právo INSERT_ANY_TABLE pro tabulky, do kterých chce uživatel vkládat data
- ❑ Právo DELETE_ANY_TABLE pro tabulky, do kterých chce uživatel vkládat data při použití parametru Repolace nebo Truncate, které slouží pro vymazání starých dat z tabulky před vložením nových



OBRÁZEK 8: PROFILER

5.9.4. Profiler

Nástroj Profiler nabízí online monitorování právě vykonávaných dotazů v databázi spolu s jejich statistikami pro vybrané sessions. Dále nabízí zvýraznění znakových řetězců pro jejich snadnější vyhledání ve výpisu, procházení výsledků již uskutečněných monitorování a spuštění nástroje TKProf na vybrané sessions.

Omezení

- ☐ připojení k databázi **Oracle Database 10g**
- ☐ **Oracle Database Server** by měl být nainstalovaný na OS řady **Microsoft Windows®** z důvodu sdílení adresáře \udump (použití Samby v Linuxu není vyzkoušené)
- ☐ práva DBA pro spuštění balíčku **DBMS_Mmonitor**
- ☐ práva pro čtení trace souborů z adresáře /udump vytvářených na straně serveru

V následujících řádkách je popsán postup pro online monitorování, které je realizováno záložkou Watching.

Před začátkem monitorování je nezbytné u vzdáleného klienta správně namapovat cestu k adresáři \udump kde se vytváří trace soubory. Tyto soubory se vytvářejí na stroji s databázovým serverem **Oracle** a jsou čteny aplikací **OracleVisualTools**. U „newindowsovských“ hostitelských OS to může být jistá komplikace.

Jestliže je aplikace nainstalovaná na stejném počítači jako server, lze tento adresář nastavit tlačítkem Default. To zajistí výběr cesty podle příslušných hodnot uložených v databázi.

Je-li však aplikace nainstalovaná na klientské stanici na síti, musí se vyřešit otázka namapování adresáře \udump. Obecně se musí tento adresář na straně serveru nastavit ke sdílení s právy pro čtení a na klientské stanici se musí namapovat jako síťová jednotka. Poté musí uživatel vybrat adresář /udump pomocí tlačítka Browse...

Jestliže došlo k modifikaci cesty k adresáři /udump, je nová cesta uložena do souboru settings.txt a při dalším spuštění aplikace **OracleVisualTools** se již zobrazí poslední zvolená cesta.

Používání **Profileru** není primárně určeno pro běžného uživatele ale zejména pro administrátory databáze nebo výkonné programátory a těm by administrátor sítě mohl dát potřebná práva, jelikož se u nich očekává znalost a zodpovědnost. Bez těchto práv tento nástroj fungovat ze stanice na síti nemůže.

Pro start monitorování sessions je nejdříve nutné tyto sessions vybrat a to tlačítkem Select Sessions. Pro výběr těchto sessions je nutné připojení k databázi. Pokud je tato podmínka splněna, zobrazí se formulář se seznamem sessions, kde každý řádek odpovídá jedné session a sloupce jsou jejich parametry. Názvy sloupců v okně odpovídají názvům sloupců v databázi:

- ☐ Session ID
- ☐ Serial ID
- ☐ User
- ☐ Schema
- ☐ Username
- ☐ Schemaname
- ☐ OSUser
- ☐ Program
- ☐ Module
- ☐ Action
- ☐ Client Identifier
- ☐ Service Name
- ☐ Type- Session type- USER/BACKGROUND

Jelikož sessions stále vznikají a zanikají, má uživatel možnost obnovit výpis sessions stisknutím tlačítka Refresh. Tlačítkem Select All lze zaškrtnout všechny sessions a tlačítkem Cancel se vrátit zpět na hlavní formulář bez přidání jakékoliv session.

Pokud uživatel vybere jednu nebo více sessions a stiskne OK, tyto sessions se zobrazí v seznamu Sessions to trace a uživatel může v průběhu monitorování sledovat stav, ve kterém se nacházejí- monitoring nebo paused. Tyto stavy jsou popsány níže.

S těmito sessions může uživatel aktuálně pracovat- modifikovat vlastnosti, které chce sledovat, spustit monitorování, pozastavit jej nebo sessions odebrat ze seznamu aktuálně vybraných. V průběhu monitorování lze vybírat další sessions z formuláře Sessions to trace a následně spustit jejich monitorování.

Jak již bylo výše zmíněno, tlačítkem Edit se zobrazí formulář, ve kterém lze nastavit, za chce uživatel vypisovat Wait events, Binds a do jaké hloubky chce sledovat dotazy, které v databázi probíhají. Tato hloubka se nastaví pomocí parametru depth. Pokud se nastaví depth na 0 nejsou zobrazovány dotazy, které vzniknou rekurzivně. Vybráním položky All se nastaví vypisování všech poddotazů. Změny se uloží stisknutím tlačítka OK a zobrazí se v seznamu Sessions to trace, tlačítkem Cancel se veškeré změny zruší.

Dále má uživatel možnost barevně zvýraznit znakový řetězec v monitorovacím okně. Toto se nastaví pomocí formuláře Colorize Settings stisknutím stejnojmenného tlačítka. Nejdříve je nutné tento řetězec zadat a následně pomocí tlačítka Set color... vybrat barvu, kterou bude tento text odlišený od ostatního. Tlačítkem Add se řetězec uloží spolu s vybranou barvou a uživatel může pokračovat v zadávání. Jestliže uživatel zadá více řetězců, bude při příležitosti výskytu více řetězců v jedné buňce mít přednost ten řetězec, který se v seznamu nachází výš. Toto lze ovlivnit tlačítky Up, příp. Down. Vybraný řetězec lze také ze seznamu odstranit jeho označením a stisknutím tlačítka Remove. Tlačítka OK a Cancel zde mají standardní význam.

Aby se veškeré nastavení projevilo, je nutné tyto vlastnosti nastavit před začátkem monitorování konkrétní session.

V okně Sessions to trace se mimo SID, SERIAL, WAIT a BINDS lze sledovat stav session. V zásadě jsou tyto stavy dva a to paused a monitoring.

Stav paused znamená, že daná session není aktuálně monitorována. Tento stav nastává bezprostředně po výběru session před spuštěním monitorování, dále pak při přerušení monitorování tlačítkem Pause a nakonec v případě výskytu chyby. V případě chyby je uživateli zobrazeno chybové hlášení a text konkrétní session se obarví červeně. Je však možné pokusit se níže popsaným způsobem opět tuto session monitorovat.

Po výběru sessions a nastavení cesty k adresáři udump lze tedy spustit proces monitorování jedné nebo více sessions současně jejich označením a stisknutím tlačítka Start. Pokud je spuštění session úspěšné, začnou se v monitorovacím okně zobrazovat příkazy příslušné session spolu se svými statistikami.

Jestliže uživatel spustil monitorování, nenastalo žádné chybové hlášení, stav session je monitoring a přesto se to monitorovacího okna nic nevypisuje, mohlo nastat několik případů:

- Ještě nebyl do databáze odeslán žádný dotaz
- Dotazy jsou zadávány, ale nic se nezobrazuje:
 - Je špatně nastaven adresář udump, opravte cestu k tomuto adresáři
 - Nemáte oprávnění pro spuštění Oracle balíčku dbms_monitor

Dále lze kdykoliv nastavit velikost písma kliknutím na tlačítko Font Size. Změna velikosti písma se projeví jak v okně, kde probíhá online monitorování, tak i v okně pro prohlížení již uskutečněných monitorování.

Z důvodu potřeby některý text zkopírovat je možné kliknout do monitorovacího okna na příslušné místo, tento text se pak zobrazí v textboxu, ze kterého je možné jej zkopírovat. Tuto volbu však zatím nenabízí okno zobrazující historii SQL příkazů.

V průběhu monitorování se vytváří soubory, kde jsou uloženy informace, které se při monitorování uživateli zobrazují na obrazovce. Tyto informace jsou uloženy v adresáři patrného z obrázku viz. OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA. Zde se po zahájení monitorování vytvoří adresář pojmenovaný podle času, kdy se spustilo první monitorování od spuštění aplikace. Jelikož by mohl být nový soubor velmi dlouhý, je rozdělen na více menších souborů pojmenovaných [pořadové číslo].txt uložených do výše zmíněného adresáře. Tyto soubory si lze po ukončení monitorování prohlížet v záložce History. Uživatel nejdříve vybere časové razítko, které odpovídá začátku monitorování, které si chce prohlížet, a zároveň pořadové číslo souboru v adresáři, kde jsou uloženy soubory s odpovídajícími příkazy a statistikami. Zobrazí se údaje z monitorování, pohybovat se po výpisu lze pomocí tlačítek |<, <, >, >|.

Upozornění: Uživatel musí mít na paměti, že monitorováním vznikají trace soubory, které mohou mít velmi velkou velikost. Aplikace OracleVisualTools tyto soubory nemaže.

TKProf

Uživatel má poté, co provede monitorování jedné nebo více sessions možnost zobrazit obsah trace souborů těchto sessions zpracované nástrojem **TKProf**. Stačí pouze dané sessions označit, přičemž musí být ve stavu paused. Pokud nejsou, zobrazí se uživateli chybové hlášení. Obsah výsledného souboru vytvořeného nástrojem TKProf se zobrazí ve WordPadu. Zatím však není možné používat nástroj TKProf na monitorování provedené v historii. Lze si však prohlížet soubory vzniklé spuštěním nástroje TKProf v adresáři tkprof viz. OBRÁZEK 6: ADRESÁŘOVÁ STRUKTURA.

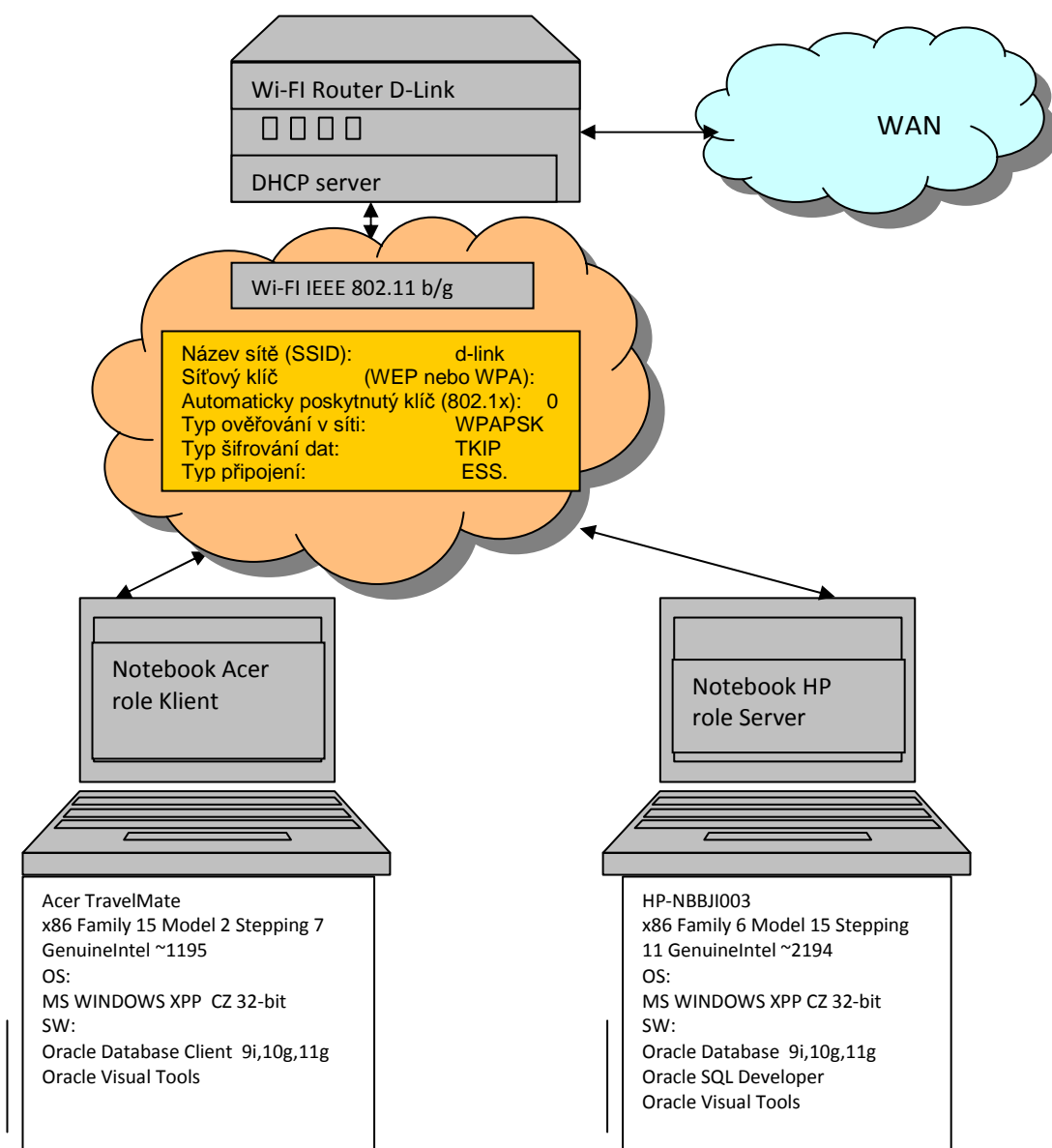
6. TESTOVÁNÍ APLIKACE

6.1. STRUČNÝ POPIS SÍŤOVÉHO PROSTŘEDÍ

Notebooky byly zapojeny do domácí sítě vytvořené bezdrátovým připojením (WI-FI) na bázi standardu IEEE 802.11 b/g
http://cs.wikipedia.org/wiki/IEEE_802.11

Oba notebooky byly zařazeny do jedné pracovní skupiny DOMA s rozsahem adres 192.168.0.1-255, IP adresy byly notebookům přidělovány DHCP serverem konfigurovaným na WI-FI routeru D-Link, síť byla typu peer-to-peer.

6.2. SCHÉMA TESTOVACÍHO PROSTŘEDÍ



OBRÁZEK 9: SCHÉMA TESTOVACÍHO PROSTŘEDÍ

6.3. POPIS INSTALACE NA HOSTITELI DATABÁZE

Notebook HP sloužil jako hostitel databázových serverů **Oracle** v téměř defaultním nastavení a konfiguraci. Každá databáze byla instalována do svého samostatného ORACLE_HOME adresáře a na samostatný port TCP konkrétně **Oracle Database 11g** verze 11.1.0 na 1521, **Oracle Database 10g XE** verze 10.2.0 na port 1522 a **Oracle Database 9i** verze 9.2.0 na port 1523.

Spuštění služeb databázových serverů a jejich listenerů bylo prováděno na počátku ručně z okna služeb. Později jsem vytvořila dávku na start i stop instancí databází.

Ukázka spuštění nejnutnějších služeb pro **Oracle Database** přes dávkový soubor start_oracle.bat

```
rem start
net start "OracleOraDb9i_homeTNSListener"
net start "OracleServiceORCL9I"
net start "OracleXETNSListener"
net start "OracleServiceXE"
net start "OracleOraDb11g_home1TNSListener"
net start "OracleServiceORCL11"
```

Ukázka zastavení nejnutnějších služeb pro Oracle database přes dávkový soubor stop_oracle.bat

```
rem stop
net stop "OracleServiceORCL11"
net stop "OracleOraDb11g_home1TNSListener"
net stop "OracleServiceXE"
net stop "OracleXETNSListener"
net stop "OracleServiceORCL9I"
net stop "OracleOraDb9i_homeTNSListener"
```

6.4. POPIS INSTALACE KLIENTŮ ORACLE DATABASE

Notebook Acer sloužil jako klientský počítač s nainstalovanými klienty **Oracle Database Client 9i**, **Oracle Database Client 10g** a **Oracle Database Client 11g**. Klienti byly nainstalováni v tzv. Custom módu s nejnutnějšími komponentami potřebnými pro funkčnost testovaného programu, tedy komponentami **Oracle Client** (obsahuje komunikační vrstvy nutné pro připojení k databázi Oracle z prostředí **Microsoft .Net Framework** verze 3.5), **Oracle Database Utilities** (obsahují spustitelné programy *exp.exe*, *imp.exe*, *sqlldr.exe*) a **Oracle Network Utility** (sloužící pro nakonfigurování a připojení klienta k databázovému serveru **Oracle** po síti – vytváří a edituje soubor tnsnames.ora). Pokud nebude provedena instalace Custom, je nutné provést administrátorskou instalaci, instalace pouze instant klienta nebo runtime nestačí, neboť neobsahuje všechny nutné komponenty pro běh aplikace **OracleVisualTools**.

6.5. DOKONFIGUROVÁNÍ INSTALACÍ

Na každém notebooku byly do přílušných ORACLE_HOME adresářů tedy %ORACLE_HOME%\network\admin nakonfigurovány soubory tnsnames.ora a u databázové instalace i listener.ora tak, aby vyhovovaly našemu multidatabázovému prostředí. Konfigurace byla provedena přímo editací souborů tnsnames.ora, pokud chyběl tak byl soubor založen.

Nakonec byla na oba počítače nainstalována aplikace **OracleVisualTools** do adresáře %Program Files% a podpora **Microsoft .Net Framework 3.5**.

Na notebooku HP v roli hostitele databázových serverů jsem navíc nainstalovala free aplikaci **OracleSQLDeveloper** verze 1.5.1, která sloužila jako zátěž databáze při testování profileru, nástroj pro nahlížení do databáze a částečně jako vodičko při vytváření grafického prostředí.

6.6. POPIS PROVÁDĚNÍ TESTŮ

Na hostitelské stanici Oracle databází jsem spustila všechny instance **Oracle Database** současně. Poté jsem na obou notebookech nakonfigurovala v **OracleVisualTools** všechny kombinace připojení do **Oracle**. Měla jsem nakonfigurováno devět připojení – ve třech ORACLE_HOME po třech připojení k databázovým instancím. Připojení bylo realizováno jak metodou Basic tak i metodou Local Naming přes TNS aliasy z souboru tnsnames.ora.

Bylo tedy nakonfigurováno devět připojení k lokálním instancím databázi a devět připojení z notebooku v roli klienta po Wi-Fi síti ke třem vzdáleným instancím databázi po domácí Wi-Fi síti typu peer-to-peer.

V této konfiguraci byly provedeny testy funkčnosti programu a to tak, že jsem procházela programem a volila různé varianty jak funkčnosti tak i parametrů. Testy se prováděly zejména k prostředí **Oracle Database 10g ExpressEdition**.

6.7. Výsledky testů

Zjistila jsem, že u **Oracle Database 9i** lze použít pro **Export**, **Import** a **SQL*Loader** jen syntaxi s TNS aliasy (metoda Basic nefunguje). U verzí **Oracle Database 10g** a výše funguje syntaxe s připojením jak Basic tak i TNS aliasů.

V programu jsem vypracovala a reálně používala jak tzv. Basic připojení tak i přes TNS aliasy ze souborů tnsnames.ora. Některé **Exporty** a **Importy** neproběhly korektně, protože možnosti **Oracle Utilities** dané adresářem ORACLE_HOME neodpovídala funkčním možnostem připojené verze databáze. Administrátor databáze musí při používání **OracleVisualTools** volit správné varianty ORACLE_HOME a verze připojené databáze. Obecně platí, že nejlepší výsledky byly při použití kombinace stejných verzí **Oracle Utilities** z ORACLE_HOME a připojení do stejné verze **Oracle Database**.

Dalším problémem byla realizace funkčnosti Profileru a jeho možnost spouštět jej z **OracleVisualTools** instalovaného na klientské stanici na síti. Tam se musela řešit otázka namapování adresáře \UDUMP s uživatelskými soubory *.trc vytvářenými balíčkem dbms_monitor na stroji s databázovým serverem a čtených aplikací **OracleVisualTools**. Obecně se musí tento adresář na hostiteli databáze dát ke

sdílení s právy pro čtení a na klientské stanici se musí namapovat tento adresář pro užití. V našem případě adresář D:\ORACLEXE\ na serveru byl ke sdílení a na klientské stanici se na něj namapovala síťová jednotka buď z prostředí **Windows Explorer** nebo příkazem net use, nejlépe v spustitelné dávce. Pak se v programu v záložce Profiler modifikovala nabídnutá cesta na skutečnou síťovou X:\product\10.1.0\admin\orcl\udump.

Samozřejmě se mohou vyskytnout nové problémy vyplývající z nasazení aplikace v jiném než testovaném prostředí, například kombinace jiných operačních systémů hostitele databáze a hostitele klienta a **OracleVisualTools**, použití češtiny v adresářové struktuře – což nedoporučuji, jiné typy procesorů – aplikaci jsem vyzkoušela výhradně na architektuře *Intel x86 32-bit*.

7. ZÁVĚR

Zadaný úkol pokrýval dvě nesourodé oblasti. Prvním úkolem bylo implementovat uživatelské rozhraní nad příkazovým řádkem pro manipulaci s daty, druhou bylo online monitorování SQL příkazů spolu se zobrazováním jejich statistik. Monitorování toho, co se děje uvnitř databáze a vylepšování výkonu database, je nikdy nekončící práce, stejně tak tato aplikace by mohla být neustále rozšiřována a vylepšována.

SEZNAM POUŽITÉ LITERATURY

- [1] **Wikipedie**
<http://cs.wikipedia.org/wiki/Datab%C3%A1ze>
- [2] **Oracle®**
<http://www.oracle.com/global/cz/database/edice.html>
- [3] **Oracle® Database SQL Developer User's Guide**
http://download.oracle.com/docs/cd/E12151_01/doc.150/e12152/toc.htm
- [4] **René Nyffeneger's collections of things on the web**
<http://www.adp-gmbh.ch/ora/concepts/awr.html>
- [5] **Oracle Base**
<http://www.oracle-base.com/articles/10g/AutomaticWorkloadRepository10g.php>
- [6] **Oracle® Enterprise Manager Concepts 10g Release 5 (10.2.0.5)**
http://download.oracle.com/docs/cd/B16240_01/doc/em.102/b31949/database_management.htm#i1006971
- [7] **Connection Strings**
<http://www.connectionstrings.com/oracle>
- [8] **Burleson Consulting**
http://www.dba-oracle.com/plsql/t_plsql_trace.htm
- [9] **Loney K., Bryla B.(2006)**
Mistrovství v Oracle 10g, Computer Press
- [10] **Oracle® DatabaseUtilities 10g Release 2 (10.2)**
http://download.oracle.com/docs/cd/B19306_01/server.102/b14215/toc.htm
- [11] **Oracle® DatabaseUtilities11g Release 1 (11.1)**
http://download.oracle.com/docs/cd/B28359_01/server.111/b28319/title.htm
- [12] **Oracle® DatabasePerformance Tuning Guide11g Release 1 (11.1)**
http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/title.htm

PŘÍLOHY

A- SEZNAM PARAMETRŮ

Módy Exportu a Importu:

Full:

Exportuje/Importuje celou databázi, pouze však uživatelem který má přidělenou roli pro full Export a full Import

Tablespace:

umožňuje privilegovanému uživateli přesouvat celé tabulkové prostory mezi databázemi

User:

Exportuje /Importuje objekty zadaného seznamu uživatelů, pokud k tomu má uživatel oprávnění

Table:

Exportuje/importuje specifikované tabulky a oddíly. Privilegovaný uživatel může kvalifikovat tabulky specifikováním schématu, který je obsahuje.

Object	Table Mode	User Mode	Full Mode	Database	Tablespace Mode
Analyze cluster	No	Yes	Yes		No
Analyze tables/statistics	Yes	Yes	Yes		Yes
Application contexts	No	No	Yes		No
Auditing information	Yes	Yes	Yes		No
B-tree, bitmap, domain function-based indexes	Yes Foot 1	Yes	Yes		Yes
Cluster definitions	No	Yes	Yes		Yes
Column and table comments	Yes	Yes	Yes		Yes
Database links	No	Yes	Yes		No
Default roles	No	No	Yes		No
Dimensions	No	Yes	Yes		No
Directory aliases	No	No	Yes		No
External tables (without data)	Yes	Yes	Yes		No
Foreign function libraries	No	Yes	Yes		No
Indexes owned by users other than table owner	Yes (Privileged users only)	Yes	Yes		Yes
Index types	No	Yes	Yes		No
Java resources and classes	No	Yes	Yes		No
Job queues	No	Yes	Yes		No
Nested table data	Yes	Yes	Yes		Yes
Object grants	Yes (Only for tables and indexes)	Yes	Yes		Yes
Object type definitions used by table	Yes	Yes	Yes		Yes
Object types	No	Yes	Yes		No
Operators	No	Yes	Yes		No
Password history	No	No	Yes		No
Postinstance actions and objects	No	No	Yes		No
Postschema procedural actions and objects	No	Yes	Yes		No
Posttable actions	Yes	Yes	Yes		Yes

<i>Object</i>	<i>Table Mode</i>	<i>User Mode</i>	<i>Full Database Mode</i>	<i>Tablespace Mode</i>
<i>Posttable procedural actions and objects</i>	Yes	Yes	Yes	Yes
<i>Preschema procedural objects and actions</i>	No	Yes	Yes	No
<i>Pretable actions</i>	Yes	Yes	Yes	Yes
<i>Pretable procedural actions</i>	Yes	Yes	Yes	Yes
<i>Private synonyms</i>	No	Yes	Yes	No
<i>Procedural objects</i>	No	Yes	Yes	No
<i>Profiles</i>	No	No	Yes	No
<i>Public synonyms</i>	No	No	Yes	No
<i>Referential integrity constraints</i>	Yes	Yes	Yes	No
<i>Refresh groups</i>	No	Yes	Yes	No
<i>Resource costs</i>	No	No	Yes	No
<i>Role grants</i>	No	No	Yes	No
<i>Roles</i>	No	No	Yes	No
<i>Rollback segment definitions</i>	No	No	Yes	No
<i>Security policies for table</i>	Yes	Yes	Yes	Yes
<i>Sequence numbers</i>	No	Yes	Yes	No
<i>Snapshot logs</i>	No	Yes	Yes	No
<i>Snapshots and materialized views</i>	No	Yes	Yes	No
<i>System privilege grants</i>	No	No	Yes	No
<i>Table constraints (primary, unique, check)</i>	Yes	Yes	Yes	Yes
<i>Table data</i>	Yes	Yes	Yes	Yes
<i>Table definitions</i>	Yes	Yes	Yes	Yes
<i>Tablespace definitions</i>	No	No	Yes	No
<i>Tablespace quotas</i>	No	No	Yes	No
<i>Triggers</i>	Yes	Yes	Yes	Yes
<i>Triggers owned by other users</i>	Yes (Privileged users only)	No	No	No
<i>User definitions</i>	No	No	Yes	No
<i>User proxies</i>	No	No	Yes	No
<i>User views</i>	No	Yes	Yes	No
<i>User-stored procedures, packages, and functions</i>	No	Yes	Yes	No

Objekty, které se objektují v příslušném módu

Zdroj : Oracle® Database Utilities 10g Release 2 (10.2) B14215-01 June 2005

Společné parametry pro Export a Import

BUFFER

Default: závislé na operačním systému

Specifikuje velikost bufferu který je použit pro přenos řádků (fetch) a to v bytech. Výsledkem je, že určuje maximální počet řádků v poli přenesených edxportem najednou.

Výpočet:

$buffer_size = rows_in_array * maximum_row_size$

Pokud je uvedena nula, Export přeneše najednou jen jeden řádek.

Tabulky s sloupci typu LOBs, LONG, BFILE, REF, ROWID, LOGICAL ROWID, or DATE jsou přenášeny po jednom řádku.

Poznámka. Používá se jen pro konvenční cestu přenosu Exportem, nemá vliv na přímou cestu Exportu

CONSTRAINTS

Default: y

Specifikuje zda jsou či nejsou exportem přenášeny omezení (constraints)

FEEDBACK

Default: 0 (zero)

Specifikuje periodu ve formě počtu řádků po kterých se má zobrazit stav zpracování.

FILE

Default: expdat.dmp

Specifikuje jména dump souborů do kterých je prováděn výstup exportu. Přípona dmp je nepovinná, můžete použít libovolnou. Export podporuje vícenosobný výstup, proto můžete použít více jmen souborů, oddělených čárkou. Názvy budou pak sekvenčně použity pro případ, že soubor dosáhne maximální povolené velikosti. Pokud bude seznam v průběhu exportu vyčerpán, budete požádáni o zadání názvu souboru.

FILESIZE

Default: Data jsou zapisována do jednoho souboru, jehož velikost je viz tabulka 19-5

Table 19-5 Maximální velikost dump souboru

Operační systém	Release Oracle Database	Maximální velikost
Any	Prior to 8.1.5	2 gigabytes
32-bit	8.1.5	2 gigabytes
64-bit	8.1.5 and later	Unlimited
32-bit with 32-bit files	Any	2 gigabytes
32-bit with 64-bit files	8.1.6 and later	Unlimited

Zdroj : Oracle® Database Utilities 10g Release 2 (10.2) B14215-01 June 2005.

Export podporuje zápis do vícenosobných souborů a Import je umí číst. Pokud specifikujete v bytech maximální velikost výstupního souboru, Export bude zapisovat soubory o této maximální velikosti. Jména exportních souborů mohou být zadána v parametru FILE, pokud ne a nebo zásoba jmen je vyčerpána pak jste požádáni zadání názvu souboru.

FULL

Default: n

Indikuje, že bude proveden úplný Export celé databáze. Uživatel musí mít přidělenou roli EXP_FULL_DATABASE. Tzv. full Export je dobrá cesta pro replikaci nebo pročištění databáze, musíme však dbát jistých pravidel, které jsou podrobně popsány v Oracle dokumentaci

GRANTS

Default: y

Specifikuje zda Export přenáší/nepřenáší přístupová práva (grants) spojená s objekty. Přístupová oprávnění jsou přenášena v závislosti na tom zda provádíme full mode Export nebo user mode Export. V full mode exportu jsou přenášena všechna přístupová oprávnění nad tabulkou v user mode jen přístupová práva poskytovaná vlastníkem tabulky. Systémová privilegia jsou přenášena vždy.

HELP

Default: none

Zobrazí popis parametrů Exportu pokud uvedeme Help=y

INDEXES

Default: y

Specifikuje, zda budou exportovány indexy

LOG

Default: none

Specifikuje název souboru do kterého bude zasílány informační a chybové zprávy

PARFILE

Default: none

Specifikuje jméno parametrického souboru obsahující seznam parametrů použitých pro Export

RECORDLENGTH

Default: závislé na operačním systému

Obsahuje délku záznamu v bytech. Parametr je nutný pro přenos výstupního souboru do jiného operačního systému používající jinou standardní hodnotu délky záznamu. Není-li hodnota zadána, použije se velikost dle použitého operačního systému.

RESUMABLE

Default: n

Parametr je použit pro aktivování/deaktivování opětné (RESUM) alokaci prostoru. Jeho nastavení souvisí s parametry RESUMABLE_NAME and RESUMABLE_TIMEOUT. Využívá se pro pozastavení exportu z důvodu nedostatku prostoru

RESUMABLE_NAME

Default: 'User USERNAME (USERID), Session SESSIONID, Instance INSTANCEID'
Hodnota tohoto parametru indikuje, který statement bude opětne alokovatelný.

RESUMABLE_TIMEOUT

Default: 7200 sekund (2 hodiny)

Hodnota parametru udává čas v sekundách, během kterých musí být chyba nedostatku prostoru vyřešena. Pokud není problém po uplynutí doby vyřešen, Export statementu je přerušen

ROWS

Default: y

Určuje zda budou/nebudou exportovány data z řádků tabulek.

TABLESPACES

Default: none

Parametr definuje, že budou exportovány všechny tabulky ze seznamu specifikovaných tabulkových prostorů včetně tabulek mající tam některý oddíl. Uživatel provádějící tento Export musí mít přidělenou roli EXP_FULL_DATABASE. Je-li parametr užit ve spojení s parametrem TRANSPORT_TABLESPACE=y můžete exportovat omezený počet tabulkových prostorů.

USERID (username/password)

Default: none

Specifikuje *username/password* (a volitelně *connect_string*) uživatele provádějícího Export. Podrobnější informace v kapitole o připojení k databázi.

VOLSIZE

Default: none

Definuje maximální počet bytů na každé výstupní pásce. Maximální hodnota je maximum, které může být uloženo do 64 bitů vašeho systému

Export- popis parametrů

COMPRESS

Default: y

Specifikuje jak Export řídí alokaci inicializačních extentů dat tabulek. Jestliže specifikujete default COMPRESS=y, pak způsob inicializace extentu je řízen importem. Pokud specifikuje COMPRESS=n , pak se zapisuje do exportu aktuální hodnoty parametrů extentů exportovaných tabulek

CONSISTENT

Default: y

Specifikuje zda Export používá klausuli SET TRANSACTION READ ONLY pro zajištění konzistence dat v čase a nedovolí jejich změnu během exportu. Parametr by se měl zadávat v případě že víme že během exportu může ke změnám docházet. Použitím consistent=n je sice tabulka přenášena v jedné transakci ale vnořené a související tabulky v jiné samostatné transakci. Pokud je tabulka rozdělena, pak každá část je exportována v samostatné transakci.

DIRECT

Default: n

Specifikuje zda je použita přímá cesta exportu. Použitím direct=y zajistíme , že data jsou exportovány přímo přeskočením vrstvy SQL command-processing . tato metoda může být mnohem rychlejší než Export konvenční cestou.

FLASHBACK_SCN

Default: none

Specifikuje tzv. system change number (SCN) Exportu pro tzv. FlashBack. Export je proveden konzistentně vzhledem k tomuto SCN

FLASHBACK_TIME

Default: none

Umožňuje specifikovat tzv. časové razítko (timestamp). Export vyhledá SCN, které nejvíce vyhovuje zadanému časovému razítku a Export je proveden konzistentně k tomuto SCN

OBJECT_CONSISTENT

Default: n

Specifikuje zda Export utilita používá/nepoužívá SET TRANSACTION READ ONLY instrukci. Jestliže je nastaven object_consistent = y, každý objekt je exportován v konzistentním stavu ve své vlastní read-only transakci, dokonce i tehdy pokud je segmentovaný. Naopak jestliže použijete parametr CONSISTENT , pak existuje jediná read-only transakce

OWNER

Default: none

Označuje že Export probíhá v user-mode a exportuje objekty dle seznamu uživatelů.

Pokud je inicializátorem exportu databázový administrátor (role DBA), může se exportovat více uživatelů

QUERY

Default: none

Umožňuje vybrat sadu záznamů z tabulek, které jsou exportovány v table-modu. Hodnota parametru je řetězec, který obsahuje SQL klausuli WHERE, jenž bude aplikována na všechny exportované tabulky. Pokud exportovaná tabulka neobsahuje sloupec z where, je signalizovaná chyba a řádky z tabulky nebudou exportovány. Při palikování parametru Query je dodržet jistá omezení uvedená v Oracle dokumentaci

STATISTICS

Default: ESTIMATE

Definuje typ databázového optimizéru statistik pokud budou exportovaná data importována. Hodnoty mohou být ESTIMATE, COMPUTE, NONE. Export bude ukládat předkalkulované statistiky do exportovaného souboru.

TABLES

Default: none

Určuje, že Export probíhá v table-mode dle zadaného seznamu názvů tabulek.

V seznamu můžeme použít SCHEMANAME, TABLENAME, PARTITION_NAME

TRANSPORT_TABLESPACE

Default: n

Jeli nastaven na y, umožňuje Export metadat tzv. přenositelných tabulových prostorů

TRIGGERS

Default: y

Určuje zda budou/nebudou exportovány databázové triggrы.

TTS_FULL_CHECK

Default: n

Je-li parametr nastaven na y, Export prověřuje zda exportovaná sada tabulkových prostorů nemá závislé objekty mimo exportovanou sadu

Import- popis parametrů

COMMIT

Default: n

Specifikuje zda by Import měl provést transakci potvrzení zápisu po každém vložení pole dat. Implicitně se provede transakce potvrzení zápisu po zavedení tabulky a Import provede rollback transakci při chybě než bude pokračovat v zavádění dalšího objektu. Pokud je tabulka rozdělena nebo obsahuje vnořené tabulky, pak se provede vložení těchto dat v samostatné transakci. Pokud nastavíme parametr na n, provede se commit transakce při každém řádku, čímž omezíme zatěžování a nadměrnému růstu rollback segmentů. Při zavádění tabulek obsahující sloupce typu LOBs, LONG, BFILE, REF, ROWID, or UROWID columns, vkládání pole se neprovádí

COMPILE

Default: y

Specifikuje se mohou provádět/neprovádět kompilace packages, procedur a funkcí po jejich vytvoření. Pokud je parametr nastaven na n, provede se kompilace až při jejich prvním spuštění

DATAFILES

Default: none

Pokud je parametr *TRANSPORT_TABLESPACE* nastaven na y, parametr udává seznam datových souborů přenášených do databáze.

DESTROY

Default: n

Specifikuje zda ano/či ne by se měly znovu použít datové soubory existující databáze. Pokud je parametr nastaven na y, Import vkládá REUSE volbu na datové soubory při vytváření tabulkových prostorů (*CREATE TABLESPACE*), čímž bude zajištěno jejich opětovné použití po předchozím vymazání jejich obsahu. Při default nastavení parametru na n bude inicializován chybový stav pokud existuje datový soubor použitý při vytváření tabulkového prostoru. Pokud potřebujete importovat do existující databáze bez nahrazení existujících datových souborů, musíte nastavit parametr *IGNORE=y*

FROMUSER

Default: none

Obsahuje seznam schémat pro Import s nimi spojených objektů, parametr je použitelný pouze pro uživatele s *IMP_FULL_DATABASE* rolí. Pokud role není přidělena pak se použije implicitně Import v uživatelském režimu a importují se objekty importujícího uživatele. Parametr se obvykle používá ve spojení s parametrem *TOUSER* kterým zadáme seznam uživatelských jmen do jejichž schémat se má Import provést. Uživatelé z tohoto seznamu již musí v cílové databázi existovat, jinak je vrácena chyba.

IGNORE

Default: n

Specifikuje, jak mají být ošetřeny chyby při vytváření objektů. Pokud použijeme implicitní parametr n. Import loguje nebo zobrazuje chyby a objekt s takto signalizovanou chybou se nenaimportuje. Pokud je parametr nastaven na y, chyby se přehlíží a nesignalizují Import objektu se provede.

INDEXFILE

Default: none

Parametr specifikuje název souboru do kterého budou ukládány SQL příkazy pro indexování, vytažené z exportního souboru. Vytvořený indexový soubor se může po importu editovat a pak ve formě SQL skriptu spustit pro vytvoření indexů. Výchozí hodnota parametru není nastavena a je použitelný pouze ve spojení s parametry *FULL=y*, *FROMUSER*, *TOUSER*, nebo *TABLES*.

SHOW

Default: n

Pokud je nastaven na y, pak SQL příkazy z exportního souboru jsou prolistovány a zobrazeny, nikoli ale objekty nainportovány. Je použitelný pouze ve spojení s parametry *FULL=y*, *FROMUSER*, *TOUSER*, nebo *TABLES*.

SKIP_UNUSABLE_INDEXES

Default: hodnota databázového parametru *SKIP_UNUSABLE_INDEXES* uvedená v inicializačním parametrickém souboru

Nastavením hodnoty parametru se přepíše nastavení z inicializačního parametrického souboru. Pokud ponese hodnotu y, vytváření indexu nad vloženým řádkem jenž je ve stavu Index Unusable state bude přeskočeno, ostatní indexy budou při vládání řádků aktualizovány. Pokud nebude parametr nastaven, vkládání řádku s indexem ve stavu Index Unusable state selže.

STATISTICS

Default: ALWAYS

Nastavuje co se děje se statistikami při importu. Může nabývat hodnot:

- *ALWAYS* - importuje statistiky
- *NONE* - neimportuje ani nepřečítá.
- *SAFE* - importuje statistiku pokud není podezřelá, jinak by se měla přepočítat.
- *RECALCULATE* - při importu přepočítává statistiky, neplatí pokud byl při exportu použit parametr *STATISTICS=NONE*

STREAMS_CONFIGURATION

Default: y

Specifikuje zda budou/nebudou z exportního souboru importována obecná metadata datových toků Stream

STREAMS_INSTANTIATION

Default: n

Specifikuje zda budou/nebudou z exportního souboru importována instancovaná metadata datových toků Stream *TABLES*

Default: none

Specifikuje, že Import probíhá v table-mode a definuje seznam jmen tabulek a jejich oddílů. Není přednastavena hodnota. Můžete specifikovat následující hodnoty parametrů:

- Jméno_tabulky - pokud nespecifikujete jméno_oddílu, pak se importuje celá tabulka včetně všech oddílů
- Jméno_oddílu - pokud uživatel uvede, bude importován jen vyjmenovaný oddíl

Syntaxe: jmeno_tabulky:nazev_oddilu

TOID_NOVALIDATE

Default: none

Nastavením na y se potlačí kontrola importního souboru na existenci typových ID (*TOID*) v aktuální databázi.

TOUSER

Default: none

Určuje seznam uživatelů jejichž schemata budou příjemcem importu. Uvedením víceprvkového seznamu v *TOUSER* musíte zajistit správné spárování s uživateli v parametru *FROMUSER*. Uživatelé v seznamu *TOUSER* se mohou opakovat.

TTS_OWNERS

Default: none

Je-li specifikován parametr *TRANSPORT_TABLESPACE=y* použijte parametr pro uvedení seznamu vlastníků dat v sadě přenositelných tabulkových prostorů

SQL*Loader- popis parametrů

BAD (bad file)

Default: jméno datového souboru s příponou bad

Parametr specifikuje jméno chybového souboru vytvářeného SQL*Loaderem pro ukládání záznamů s chybami v procesu vkládání dat. Není-li uveden, použije se implicitní jméno s příponou bad spojené s prvním INFILE

BINDSIZE (maximum size)

Default: zjistí se spuštěním sqldr bez uvedení parametrů

Specifikuje maximální velikost BIND pole.

COLUMNARRAYROWS

Default: zjistí se spuštěním sqldr bez uvedení parametrů

Určuje počet řádků alokovaných pro v módu pro přímou cestu. Hodnota není počítána SQL*Loaderem, musíte ji specifikovat nebo použít implicitní..

CONTROL (control file)

Default: none

Specifikuje jméno řídicího souboru SQL*Loaderu který popisuje jak mají být data zaváděna. Přípona souboru není specifikována, implicitně je .ctl. není jméno souboru uvedeno , SQL*Loader o něj požádá

DATA (datafile)

Default: jméno řídicího souboru s příponou .dat.

Specifikuje jméno datového souboru obsahující vkládaná data. Není-li uveden použije soubor .dat. Soubor uvedený na příkazové řádce je zaveden jako první a první soubor INFILE z parametrického souboru je ignorován. Další datové soupy z parametrického souboru jsou zpracovány.

DATE_CACHE

Default: Enabled (pro 1000 elements)., pro vypnutá datové vyrovnávací paměti uveďte 0.

DATE_CACHE specifikuje velikost datové vyrovnávací paměti v počtech vstupů. Parametr je dostupný pouze pro přímou cestu vkládání. Pokud počet vstupů přesáhne default nebo zadanou hodnotu, je vyrovnávací paměť automaticky vypnuta pro tuto tabulku

DIRECT (data path)

Default: false

Specifikuje přímou cestu zavádění pokud je true, false specifikuje konvenční cestu zavádění.

DISCARD (filename)

Default: jméno datového souboru s příponou .dsc.

DISCARD specifikuje discard soubor (volitelně) vytvářené SQL*Loaderem pro ukládání záznamů které nebyly vloženy nebo tabulkou odmítnuty..

DISCARDMAX (integer)

Default: ALL

DISCARDMAX specifikuje počet odmítnutých záznamů než bude zavedení ukončeno. Pro zastavení zavádění dat na prvním odmítnutém záznamu, zadejte 1.

ERRORS (errors to allow)

Default: zjistí se spuštěním sqlldr bez uvedení parametrů

ERRORS udává maximální počet chyb při vkládání. Je-li tento počet překročen proces vkládání dat je ukončen

EXTERNAL_TABLE

Default: NOT_USED

EXTERNAL_TABLE určuje zda bude SQL*Loader zavádět data užitím volby external table. Má tyto možné hodnoty:

- NOT_USED - implicitní hodnota, zavedení dat bude provedeno v modu konvenční nebo přímou cestou.
- GENERATE_ONLY – umístí všechny SQL příkazy potřebné pro zavedení dat přes externí tabulky do log souboru. Tyto záznamy mohou být následně upraveny. Zavedení dat pak může být provedeno později bez použití SQL*Loaderu spuštěním vygenerovaného sql skriptu v SQL*Plus.
- EXECUTE – povolí provedení SQL příkazů potřebných pro zavedení dat použitím externích tabulek.. Při chybě ve vykonání SQL příkazu bude zavádění dat zastaveno.

-

FILE (file to load into)

Default: none

FILE specifikuje soubory pro databázové extenty, Má význam jen při paralelním zavádění dat.

LOAD (records to load)

Default: All records are loaded.

LOAD specifikuje maximální povolený počet logických záznamů k zavedení

LOG (log file)

Default: jméno řídicího souboru s příponou .log.

LOG specifikuje log soubory které t SQL*Loader bude vytvářet pro ukládání logovacích informací v průběhu zavádění dat

MULTITHREADING

Default: true u víceprocesorových systémů, false na jednoprocessorovém systému. Je dostupný pouze pro přímou cestu zavedení dat.

PARALLEL (parallel load)

Default: false

PARALLEL specifikuje zda v modu přímé cesty zavedení dat může být nakládáno s více souběžnými sessions pro zavedení dat do stejné tabulky

PARFILE (parameter file)

Default: none

PARFILE specifikuje jméno souboru s společně použitelnými parametry

READSIZE (read buffer size)

Default: zjistí se spuštěním sqlldr bez uvedení parametrů

The *READSIZE* parameter je užit jen při čtení dat z datového souboru. Při čtení záznamů z řídicího souboru je použita vždy hodnota 64 KB jako READSIZE. Specifikuje v bytech velikost čtecího bufferu, nechcete-li použít implicitní hodnotu. Maximální hodnota je 20 megabytes (MB)

RESUMABLE

Default: false

RESUMABLE parametr je užít pro zapnutí a vypnutí resumable alokaci souborů. Pokud použijete parametry *RESUMABLE_NAME* a *RESUMABLE_TIMEOUT* musíte parametr nastavit na true

RESUMABLE_NAME

Default: 'User USERNAME (USERID), Session SESSIONID, Instance INSTANCEID'

Hodnota specifikuje řetězec s názvy statamentů které budou použity s volbou resumable. Pro specifikaci statementu můžete použít pohled *USER_RESUMABLE* nebo *DBA_RESUMABLE*. Parametr je ignorován není-li použito *RESUMABLE* true

RESUMABLE_TIMEOUT

Default: 7200 seconds (2 hours)

Hodnota parametru udává čas v sekundách, během kterých musí být chyba nedostatku prostoru vyřešena. Pokud není problém po uplynutí doby vyřešen, proces zavádění dat je ukončen

Parametr je ignorován není-li použito *RESUMABLE* true

ROWS (rows per commit)

Default: zjistí se spuštěním sqlldr bez uvedení parametrů .

Při konvenční metodě zavádění dat specifikuje počet řádků v bind poli. Při přímé metodě zavedení dat specifikuje počet řádků které chcete číst z datového souboru před jejich uložením. Implicitně jsou čteny všechny záznamy a ukládány až do konce

SILENT (feedback mode)

V proběhu provádění SQL*Loaderu jsou zobrazovány na obrazovce různé informace a zprávy, které se ukládají také do log souborů

Parametr *SILENT* umožňuje potlačit zobrazované informace

Lze použít následujících hodnoty parametru::

- *HEADER* – nezobrazuje hlavičky zpráv, ne však v logu.
- *FEEDBACK* – nezobrazuje zprávy typu "commit point reached".
- *ERRORS* – nezobrazuje chybové zprávy .
- *DISCARDS* – nezobrazuje zprávy o zaznamech zaznamenávaných do discard souboru (záznamy, vyřazené vstupním filtrem)
- *PARTITIONS* - Vypne zápis per-partition statistik do log souboru při přímém modu zavádění dat
- *ALL* – implementuje funkčnost hodnot: *HEADER*, *FEEDBACK*, *ERRORS*,
- *DISCARDS*, and *PARTITIONS*.

SKIP (records to skip)

Default: žádný záznam není vynechán

SKIP specifikuje počet logických záznamů od začátku souboru, které nebudou zavedeny.

SKIP_INDEX_MAINTENANCE

Default: false

SKIP_INDEX_MAINTENANCE parametr zastavuje udržbu indexů pro přímý mód zavádění ale nemá vliv na konvenční mód . Nastavuje příznak Index Unusable.

SKIP_UNUSABLE_INDEXES

Default: hodnota konfiguračního parametru Oracle databáze *SKIP_UNUSABLE_INDEXES*

Hodnota zadaná v SQL*Loader přepisuje hodnotu konfiguračního parametru. Hodnota true znamená že zavádění dat pokračuje když se objeví příznak Index Unusable na používaném indexu.

STREAMSIZE

Specifikuje velikost v bytech pro datové proudy při přímém modu zavádění

USERID (username/password)

Default: none

Specifikuje uživatele a jeho heslo, Pokud se připojujete jako SYS musíte uvést také AS SYSDBA